UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
School of Aeronautical and Astronautical Engineering

Design and Optimisation of an Uncrewed Aerial System Service Framework

by

Robert Entwistle

A thesis for the degree of
Doctor of Philosophy

January 2023

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Faculty of Engineering and Physical Sciences
School of Aeronautical and Astronautical Engineering

Doctor of Philosophy

Design and Optimisation of an Uncrewed Aerial System Service Framework

by Robert Entwistle

Creating an effective and competitive uncrewed aerial system (UAS) service requires a large array of decisions to be made. The solution has to encompass not only the choice of ucrewed aerial vehicle (UAV), but also the concept of operations, the location of the operating base(s), the personnel required to fly and maintain the platforms and the effect of the temporal weather to name a few. Currently, the decisions are made based on little evidence and previous knowledge of successes and failures. Here, the creation and use of a simulation tool to aid the decision making process of designing a UAS service is investigated.

This thesis introduces a mission-based simulation tool that utilises discrete-event simulation techniques to replicate a real-world UAS service proposal. For the given service the tool models the UASs in terms of performance and reliability, and places them at operating bases with the required personnel. Missions and weather variables are dynamically generated from predefined probability-distribution functions set out in the service proposal. The simulation ultimately produces a score that signifies the effectiveness of the service design along with the cost. With these outputs and the data behind them, a design-value is produced from a value function. By running the simulation with different design candidates consisting of different combinations and numbers of UAV types at different operation bases it is possible to find an optimal service design. This is demonstrated in a case study applied to the simulation tool.

The general lessons learnt while developing the computational tool are discussed and the model's scalability and applicability was explored. The presented tool is capable of modelling a multitude of UAS service types and mission profiles. The framework is based on a modular, comprehensible, generic and realistic approach which benefits the applicability of the tool and the ability to update components. To perform large optimisation studies the addition of a combinatorial problem solver is recommended and discussed as future work.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ABM	Agent Based Modelling
ABS	Agent-Based Simulation
AHP	Analytical Hierarchy Process
AOA	Angle Of Attack
API	Application Programming Interface
BAU	Business as Usual
BLOS	Beyond Line of Sight
CAA	Civil Aviation Authority
CBA	Cost Benefit Analysis
CBS	Cost Breakdown Structure
CLI	Command Line Interface
CoG	Centre of Gravity
CONOPS	Concept of Operations
COTS	Commercial Off The Shelf
CVRP	Capacitated Vehicle Routing Problem
DEM	Discrete Event Modelling
DES	Discrete Event Simulation
DVRP	Dynamic VRP
EO	Electro-Optical
FIFO	First In First Out
FOV	Field of View
FPA	Focal Plane Array
FSM	Fleet Size and Mix
GCS	Ground Control Station
GIQE	General Image Quality Equation

GIS	Geographic Information System
GSD	Ground Sample Distance
GUI	Graphical User Interface
HALE	High Altitude Long Endurance
HFVRP	Heterogeneous or mixed Fleet VRP
HTML	Hyper Text Markup Language
HVRP	Heterogeneous VRP
ICE	Internal Combustion Engine
ISR	Intelligence, Surveillance and Reconnaissance
LOS	Line of Sight
MADM	Multi-Attribute Decision Methodology
MALE	Medium Altitude Long Endurance
MAUT	Multi-Attribute Utility Theory
MCA	Maritime Coastguard Agency
MCS	Monte Carlo Simulation
NASA	National Aeronautics and Space Administration
NIIRS	National Imagery Interpretability Rating Scale
NPV	Net Present Value
OSD	On-Screen Display
RCM	Reliability-Centred Maintenance
RF	Radio Frequency
RFP	Request for Proposal
RNLI	Royal National Lifeboat Institution
SAR	Search and Rescue
SDK	Software Development Kit
SFC	Specific Fuel Consumption
STK	Software Tool Kit
UAS	Uncrewed Aerial System
UAV	Uncrewed Aerial Vehicle
VCD	Value-Centric Design
VCDM	Value-Centric Design Methodology
VRP	Vehicle Routing Problem
VTOL	Vertical Take-Off and Landing

Declaration of Authorship

I, Robert Entwistle, declare that this thesis entitled Design and Optimisation of an Uncrewed Aerial System Service Framework and the work presented in it are my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. None of this work has been published before submission.

Signed:..... Date:.....

Acknowledgements

Firstly, I would like to acknowledge the funding bodies that have made this research possible. This work has received financial support from the EPSRC Centre for Doctoral Training in Next Generation Computational Modelling grant EP/L015382/1. It is also part sponsored by QinetiQ as an industrial sponsor of the NGCM CDT.

I would like to thank my supervisor, Prof. Jim Scanlan for his continuous support, invaluable advice and patience throughout all stages of this research. My thanks also extends to all my colleagues and friends for their kind help and for all the experiences gained through these relationships. Finally I would like to express my gratitude to my parents Clive and Annie, my amazing wife Lauren, and finally my beautiful daughter Erin. Without their tremendous understanding and encouragement in the past few years, it would have been impossible for me to complete my study.

Chapter 1

Introduction

Due to the increasing capability and availability combined with the decreasing cost of uncrewed aerial systems (UASs), both the commercial and military UAS markets are expected to continue to boom. At the beginning of this research some reports expect the worldwide market to increase from $6.8B in 2016 and reach $36.9B by 2022 [1]. Now, in 2021 the overall UAS market value is estimated to be $27.4B with a new prediction of $58.4B in 2026 [2]. Therefore, for companies under this market umbrella, it is increasingly important to offer a product that will stay ahead of the competition. In addition to these worldwide market values, some market forecasts go on to predict the breakdown of the market. It is noted that a significant proportion of the small-UAS market will be driven by the sales of commercial applications and services as opposed to just the sales of the hardware [3]. Service providers are seen as one of the four key stakeholders of the UAS market ecosystem alongside platform manufacturers, subsystem manufacturers and software providers [2].

The applications and services of UASs in the military context range from intelligence, surveillance and reconnaissance (ISR) assignments to weapon deployment as well as peace-keeping missions. Military applications were the developing ground for the early UAS. In 1915 Nikola Tesla had visions that wireless control of aeroplanes at a distance could be used for attack as well as defence [4]. This was followed up by A. M. Low's 'Aerial Target', built and tested between 1916 - 1917 which was designed to be a remotely controlled aerial torpedo and is widely regarded as the first uncrewed aerial vehicle (UAV) [5].

In the commercial, consumer, civil and governmental areas, the most recent surge of development came from merging hobbyist remote-control model-aircraft with the advances in the fields of light-weight sensors and electronics, and control algorithms. Now, there is a vast variety of UAS platforms and an increasing number of applications in these areas, for example: monitoring agriculture, assisting with search and rescue (SAR), surveillance, photography, videography, providing aerial surveys and maintenance

inspections for buildings and structures as well as supporting the emergency services. In fact, incredibly, the majority of these applications were foreseen in 1898 by Nikola Tesla who filed a patent which covered wireless controlled vessels (before the first flight) with the following suggested applications: carrying letters or packages; exploring and establishing communication with inaccessible regions; other scientific, engineering or commercial purposes; and warfare [6].

By focusing on the UAS service as a whole, it can be treated as a product in its own right. This is to say that all the major design aspects of the service need to be carefully addressed to ensure it achieves the goals originally proposed. For example, which UAV to use and how many of them or if a combination of different UAVs would give a better result and where should they be located. Requirements set by all the stakeholders need to be taken into account to find the optimal service design. Thus, the task of designing the service becomes a multi-stakeholder, multi-disciplinary problem.

In the context of this thesis, a UAS service is considered to be all that is required to get a UAS airborne and complete its set of tasks. The tasks are a treated as repetitive events, but not necessarily structured or consistent in terms of frequency and duration. This short definition is expanded further in Section 1.1. To help picture this in the meantime, a service for monitoring the rainforest for wildfires is a simple example. This service could consist of tasks that include daily surveys of the region but also tasks that need to respond to fires by monitoring and tracking them over a period of time.

To design a service such that it is effective at completing its tasks, both physical or virtual models and methods can be used to find an optimal solution. A physical, iterative (or even trial and error) approach can be used to confirm or refute decisions and refine the service. However, this technique can be expensive, time consuming or impractical, due to the expected cost of the service, time constraints, or the size of the service. In contrast, the use of high-powered computers and mathematical models have allowed other complex engineering design and optimisation problems to be explored relatively cheaply and quickly [7]. Either static or dynamic computational models of the UAS service can be created to represent the problem and simulated to find an optimal design solution. Nevertheless, it is worth noting that, as quoted in Neumann [8], 'simulations are only as good as the assumptions on which they are based', with Neumann adding 'in fact, they may not even be *that* good'.

With this in mind, Wiese & John [9] state that a generic engineering design process should encompass:

- A systematic approach in the presentation of the problem and the way potential solutions are proposed and evaluated.
- An iterative approach via the use of simulations and prototyping to expose the characteristics of the solutions.

- A multi-disciplinary approach since several different disciplines are involved in the decision process.

These best practices should be seen as fundamental practices to all multidisciplinary design processes and be employed where possible.

1.1 Uncrewed aerial system service

As briefly introduced earlier, in the context of this project, a UAS service is considered to be all that is required to get a UAS airborne and complete its task. This encompasses the aspects listed below and has been visualised in Figure 1.1. This list is not exhaustive but gives a good overview of the building blocks to a service.

1. The UAS equipment. This includes not only the airframe itself, but also all ground systems such as the ground control station (GCS); launch and recovery systems; and ground-based communication and data relay systems for data exploitation. It also includes all air systems such as the payload and air-based communication systems.

2. The facilities. This comprises of personnel accommodation; power and amenities; tools and equipment; storage and hangarage; and physical security if required.

3. Consumables. This covers the supply of fuels, batteries and lubricants. Consumables require either the correct amount for the entire service to be supplied at the beginning or to top-up supplies when resources run low.

4. Personnel: both operational and maintenance personnel. This incorporates pilots, technicians, data-analysts, security and managers, all of whom require training (and, in some roles, qualifications), salaries and personal equipment.

5. Insurance and legal. This includes all paperwork to ensure the flights are conducted within the laws of the area they are in, and with the correct insurance for all aspects of the service, such as aviation insurance and medical insurance.

6. The concept of operations (CONOPS). This includes the mission profiles; how much flying is required, including when, where and the availability of the UAS; how the payload is used; and how the information is collected, distributed and analysed.

Therefore, it can be seen that a service is much more than just an airframe alone. There are additional physical, virtual, consumable, and human resources required to make the airframe perform a useful task and hence provide a service. This is what is considered, in this project, to be the backbone of an uncrewed aerial system service.

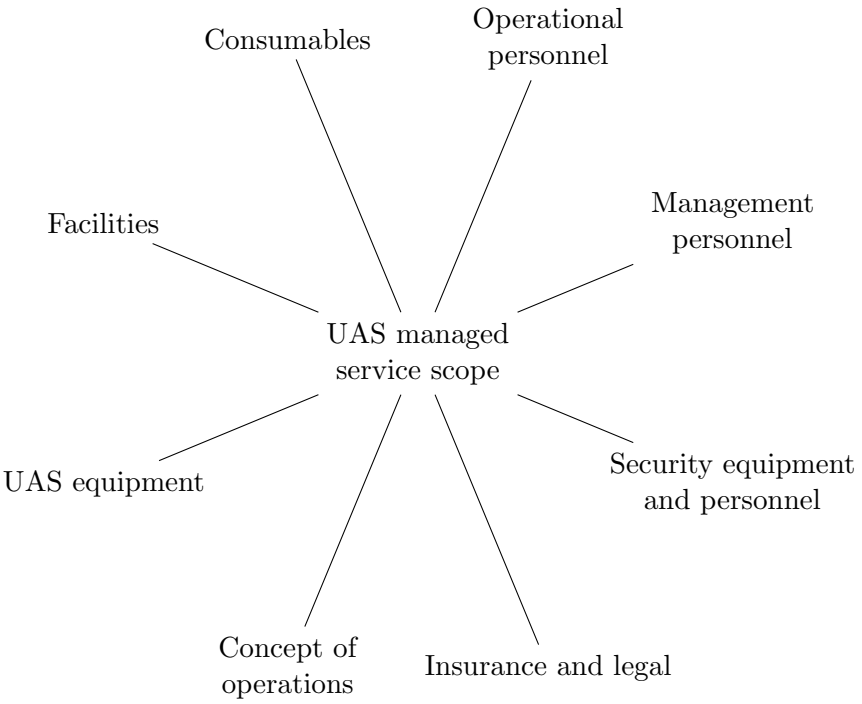


FIGURE 1.1: Mind map showing the scope of the Uncrewed Aerial System (UAS) service. Each node has further sub-nodes that have not been shown here but are discussed in Section 1.1. This mind map is adapted from QinetiQ's *Uncrewed Aerial System (UAS) Managed Service Price Model* document.

1.2 Motivation

The work presented in this document is towards the creation a computational tool that aims to aid the design of a UAS service based on the requirements set by the client. The service solution should put the right payload(s) in the right place(s) to deliver the right information at the right time(s) to the right people at a competitive and realistic price. Most decisions that are required to be made throughout the design process are complex trade-offs between multiple parameters resulting in a potentially sensitive design space: a small change in the inputs could have a large effect on the outputs. This creates a challenging design and optimisation problem which falls into both the *connectivity* sub-type of structural complexity (associated with difficulty deconstructing causes and effects) and the *social and political* sub-type (associated with multiple objectives and multiple stakeholders) [10]. In this study the UAS service can be considered as the sole logistics system. However, the UAS service may well be integrated into a larger service, for example providing additional searching capabilities to a search and rescue service.

This leads to the problem of comparing design solutions across multiple attributes to ascertain which is "better" because "better" is difficult to define formally. However, the application of the value-centric design framework aims to overcome this challenge. Instead of setting requirements to attributes (such as a weight requirement), the value-driven design framework assigns an objective function that converts the full set of attributes

into a score. Then it is the design team's task to find or create a design that yields the highest score [11]. To apply this methodology to the design of a UAS service the design candidates need to be modelled. For example, the ability of the UAS to fly in adverse weather conditions might have a high weighting on the objective function to ensure minimal service disruption.

Presently, this problem is generally tackled in a non-qualitative, non-quantitative manner through the adaptation of previous service designs. Therefore it is challenging to provide evidence that the service solution meets the aims set by the client in the request for proposal (RFP). Instead, by using a computational model of the service it should be possible to gain a quantitative insight to the composition of the service cost and how successful the design candidate is at performing the service, thus allowing an optimal design to emerge from the pool of candidate designs.

1.3 Research aim and objectives

Based on the motivation presented above in Section 1.2, this research aims to

> *explore the development and application of a mission-based computational simulation and optimisation environment to have transformational impact on decision-making when designing an uncrewed aerial system service.*

The decision-making element of the service design presented in this work predominately relates to the choice of operating-base locations and facilities, the choice of UASs and related equipment (including asset heterogeneity and operational consumables), and the number of personnel involved in the operation. This covers the technical and operational elements presented in Figure 1.1, whereas the non-technical and legal components are out of scope for this work.

This research aim can be broken down into the following objectives. These can be seen as the path this thesis will explore:

1. to investigate, define and model a UAS and its performance;
2. to develop a computational model and simulation environment to optimise UAS deployment as a service;
3. to quantify the capabilities of the model through application to a specific case study in the Solent region;
4. to assess the applicability and scalability of the model to support decision-making when designing a UAS service.

The following research questions were formulated from the identified research objectives. These questions will be central throughout this document:

1. what are the capabilities, critical areas and key findings when applying the model and simulation to a case study?
2. how applicable and scalable is the model for UAS service design?

1.4 Document structure

The remainder of this document consists of 6 chapters which address the following content:

- Chapter 2 (Literature review) presents the literature review on the main themes throughout this project. This includes the definition and categorisation of UAS and modelling techniques for the performance of the UAS and its payload. It also covers an overview of simulation and modelling techniques and some best practices. Finally, it reviews the use of value-centric design methodologies to capture the value of the system and the stakeholders' needs.
- Chapter 3 (Developing a UAS decision support framework) contains details on the framework of the computational model developed within this report. The requirements and design of the framework are presented along with the assumptions, simplifications and limitations in the model and this thesis. These are provided with justification and reasoning and the effect they have on the model and its output is discussed.
- Chapter 4 (Simulation and modelling) provides details on the simulation and modelling methods applied. The choice of tools to achieve the requirements of the model set out in the framework discussion are presented. The selected tool is identified and the method the UAS service is modelled is explained. This discusses the flow of the missions through the model's process and how the UAS and associated resources respond to the events.
- Chapter 5 (Results and analysis) presents a case study to demonstrate how the framework models a service and the type of analysis the framework allows the service designers to perform. The service inputs, including the mission descriptions, the operating base locations, the UAS platforms modelled and the service policies applied are presented and the results are analysed. The findings from the simulation are discussed, highlighting the benefits and shortcomings of the model.

- Finally, Chapter 6 (Discussion and conclusions) addresses the broader issues around the tool development and the key findings from this research. The model's complexity, transparency and computational measures are discussed. Alongside this, the scalability with operational size and scope, asset heterogeneity and applicability to a variety of business-models is commented on. Finally, it provides a conclusion to the work presented in this thesis. It revisits the research aims and objectives and comments on how the simulation environment has performed. Also discussed in this chapter is the future work required to further this model.

Chapter 2

Literature review

This chapter contains a review of the literature and research areas relevant to aiding the design of a UAS service. Section 2.2 presents an overview of uncrewed aerial systems, their composition and how to categorise and model them for mission based simulations. The following section, Section 2.3, provides details on modelling and simulation techniques. Some best practices are described to understand how to achieve better model confidence with respect to the level of detail. Section 2.4 reviews the design decision and value-centric design methodologies. Some common techniques used to obtain value functions and rank attributes in preference order are presented.

2.1 Examples of commercial UAS services

Section 1.1 provided the definition of a UAS service in the context of this thesis. By way of background information to aid understanding of the topic, a few commercial uncrewed aerial system services are presented below.

Zipline[1] is an example of a commercial drone-delivery service that, at the time of writing have flown more than 27,889,000 miles and completed over 392,800 commercial deliveries with 3,929,000 products delivered [12]. Zipline currently have 10 distribution centres active in 4 different countries mostly distributing medical products using an air-drop method. The distribution centres are purpose built and contain the infrastructure to launch, land, operate and maintain the UAS. The UAS, which can carry 1.8 kg of payload and can fly 300 km on a single charge, is launched using an electric catapult and lands via an aerial arresting gear. Despite the achievable range, the operations are limited to a 80 km radius to build in safety. Other safety measures include two electric motors (with the ability to fly safely on a single motor), and a ballistic recovery system (parachute) to reduce the kinetic energy of the system should a serious failure occur and flight is no

[1] See `https://www.flyzipline.com/` for more details on the company.

longer possible. At the outset of their Rwandan service, Zipline had a fleet of 15 UAS at one distribution centre that would complete between 50 to 150 on-demand emergency deliveries per day to the 21 facilities within the 80 km operation radius [13]. These operations flew pre-planned routes from the distribution centre to the known air-drop destinations and then returned.

Another example is the UK Maritime Coastguard Agency (MCA) tender for the next generation Search and Rescue (SAR) aviation programme (known as UKSAR2G) to provide His Majesty's Coastguard search and rescue helicopters, planes and remotely piloted drones [14]. Under the commercial strategy of this tender, the MCA based the structure of the tender on findings from a data model of UK SAR responses which led to three lots being allocated. Lot Three focused on '*fixed-wing and potentially UAV, rapid search only, surveillance and pollution response*'. This is the lot for UAS entries, with a very broad description of rapid deployment capabilities for searching and surveillance tasks. The key messages presented from the data modelling were that 94% of tasks occurred within 150 NM of the closest base location, and that a second asset at a base reduced the average response distance by 3 NM [15].

In this thesis the focus is on the optimisation challenge of selecting the operating base(s) and the choice of UAS(s) at each operating base. This involves considering and modelling the UAS performance and reliability, the personnel requirements, the required facilities and infrastructure, and the CONOPS (all of which were highlighted as the building blocks of a UAS service in Section 1.1). This optimisation challenge was shown to be an important element in the UKSAR2G programme's data modelling and involves the combination of several different research disciplines. The following sections delve into these disciplines and present the relevant literature and research to aid this work.

2.2 Uncrewed aerial systems

Uncrewed aerial systems are described by the UK's Civil Aviation Authority (CAA) as an '*evolutionary component of the aviation system*' [16]. They also go on to define an uncrewed aircraft in the 'Unmanned Aircraft System Operations in UK Airspace - Guidance' (CAP722) document as '*an aircraft which is intended to operate with no human pilot on board that is capable of sustained flight by aerodynamic means; is remotely piloted and/or capable of degrees of automated or autonomous operation; is reusable; and is not classified as a guided weapon or similar one-shot device designed for the delivery of munitions*' [17].

UASs are often used for dull, dirty or dangerous missions. Dull missions are, for example, extended surveillance where the lapses in concentration in a crewed flight means loss of mission effectiveness. Dirty missions include crop-spraying with toxic chemicals, and dangerous missions include military operations as well some civilian operations, such as

power-inspection and forest fire control [18]. Austin [18] believes that these three roles do not go far enough and adds three further roles to this list: covert, where is it imperative not to alert the target; research, where the use a UAV as a test-bed for novel technology can dramatically reduce costs and risks; and finally, environmentally critical roles, where using a UAV will reduce pollution or environmental disturbance (the example given in [18] is regular inspection of power-lines due to reduced noise for local residents and less disturbance to farm animals from noise and sight).

The removal of the pilot from the aircraft is advantageous in terms of operator safety and reducing the risk of dangerous missions, yet disadvantageous due to the system needing to be automated or controlled remotely [19]. Sensors and instrumentation are required to replace the senses of the on-board pilot as the human operator is still legally responsible for the aircraft [17]. The level of situational-awareness data provided from these sensors vary from aircraft to aircraft, with some providing a higher level than a crewed aircraft [20].

The cost of manufacturing a UAS (both the air vehicle and control station) for a surveillance task is approximately of the order of 40 - 80% of the cost of a crewed aircraft for the same task [18]. This comparison is mostly based on the savings made in structure and volume (and therefore mass) due to the removal or the aircrew, and the replacement of the aircrew with an electro-optical sensor. It does also include the ground control station in the costing. The operating costs are also lower than that of a crewed aircraft, Austin [18] estimates it at 40% or less overall of the crewed aircraft cost, due to lower maintenance and fuel costs[2] amongst other factors such as hangarage and crew salaries. Austin does state that this figure and those behind it are, to some degree, inevitably subjective.

2.2.1 UAS composition

The UAS is comprised of not only the aircraft but several other systems. The CAA defines an uncrewed aerial system as being comprised of '*individual system elements consisting of the uncrewed aircraft and any other system elements necessary to enable flight, such as a remote pilot station, communication link and launch and recovery element. There may be multiple uncrewed aircraft, remote pilot stations or launch and recovery elements within a UAS*' [17]. This definition is reiterated in Austin [18] and with the addition of the following elements:

- the payload - ranging from simple camera system to high-power radar systems.
- navigation systems - where the level of sophistication depends on the level of autonomy.

[2]Austin [18] puts maintenance and fuel costs at 20% and 5% of the crewed aircraft cost respectively.

- interfaces - both internally within the total system (*i.e.* interfacing the control station with the air vehicle) but also externally, for data dissemination.
- support equipment - ranging from operating and maintenance manuals to tools, spares and specialist equipment.
- transportation - which needs to be provided for all the sub-systems mentioned above and depends on the size and operation of the UAS.

Figure 2.1 depicts the functional structure of a UAS as described by Austin [18].

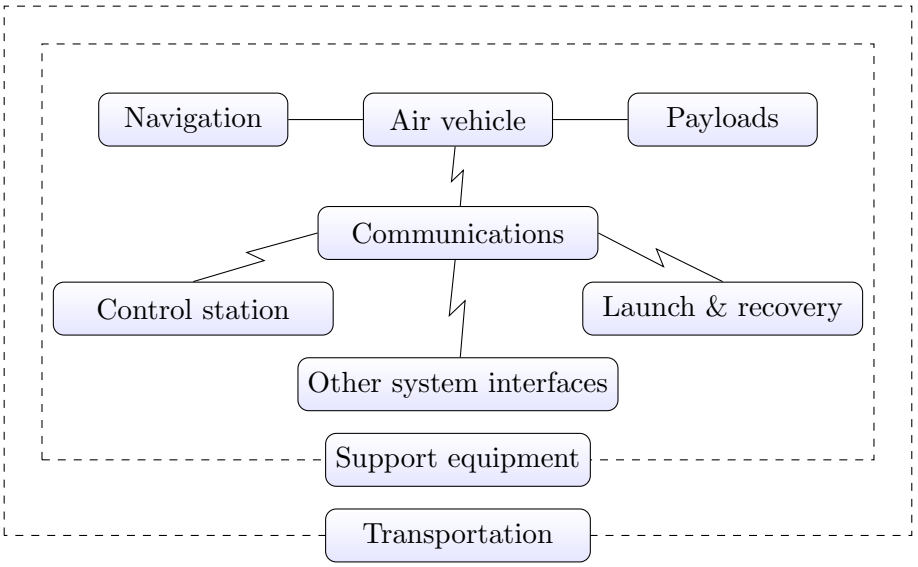


FIGURE 2.1: Functional structure of an unmanned aerial system. Reproduced from Austin [18]. Solid-straight lines denote physical connections, whereas solid-zig-zag lines denote other types of connections. The dashed boxes encapsulate those items affected by that item.

2.2.2 UAS categorisation

UASs have many features by which they have been categorised (fixed wing and rotary wing, powered and unpowered, lighter-than-air and heavier-than-air, hand-launched and runway-dependant, to name a few). However, the most common categorisation is based on the capability or size of the air vehicle although the boundaries can be rather vague. The categories are as follows [18]:

HALE - high altitude long endurance. These operate at altitudes over 15,000 m with an endurance over 24 hours. Common operations are extremely long range ISR and they are runway dependant. An example is the Northrop Grumman RQ-4 Global Hawk with its 39.9 m wingspan and nearly 15,000 kg gross weight [21] (see Figure 2.2a).

MALE - medium altitude long endurance. These often have an endurance of around 24 hours and operate at 5,000 - 15,000 m altitude and offer a similar capability to HALE just with a shorter range. An example is the Israel Aerospace Industries Heron with a line of sight (LOS) range of approximately 350 km and a beyond line of sight (BLOS) range of approximately 1000 km [22] (see Figure 2.2b).

Tactical UAV. These typically have a range of between 100 and 300 km and offer simpler capabilities than MALE and HALE. They have an endurance of normally less than 24 hours. An example is the Thales Watchkeeper WK450 which can achieve a LOS range in the region of 140 km [23] (see Figure 2.2c).

Light UAV. The entries above this category are dominated by military, peace-keeping or governmental UAVs. This category, according to the CAA CAP722 [17], contains UAVs that provide diverse civilian operations such as cinematic aerial filming, crop spraying and research. An example is the University of Southampton's Spotter Light UAS with a maximum take-off weight of 35 kg [24] (see Figure 2.2d).

Small UAV. This category is heavily populated with civilian recreational UAVs and is often defined as less than 25 kg gross mass. This civilian presence is most likely due to more attainable flying permissions for operators and therefore a larger market audience. Small UAVs also have a presence in military applications due to their portability. An example of a small UAV is Sulsa (Southampton University Laser Sintered Aircraft) with a take-off weight of less than 4 kg [25] (see Figure 2.2e).

Micro UAV. The final category is for UAVs that are operated in urban environments, particularly within buildings [18]. These generally have an endurance of less than one hour and fall into the rotary or flapping wing category of UAV due to the required manoeuvrability in these environments. An example is the sub 33 gram FLIR Black Hornet PRS [26] (see Figure 2.2f).

Both rotary and fixed-wing UAVs can be classified into these categories. However, it should be noted that there is an apparent correlation between size and lifting-method, such that as the size increases, the amount of fixed-wing UAVs compared to rotary-wing increases. This is mostly due to the need for manoeuvrability at the small scale which is offered by rotary-wing, and endurance at the large scale which is offered by fixed-wing.

Recently, at the end of 2020, the UK's CAA moved away from regulating UAVs mostly by mass limits and now include regulations that are proportionate to the level of risk that the operation presents along side mass limits [17]. In doing so, they increased the mass limits in the lower category (now defined as the Open Category) from 20 kg (not including fuel) to 25 kg (maximum take-off mass) [27].

Throughout this thesis, the focus is set on fixed-wing aircraft that fall into the *small* and *light* UAS categories. This is because the service types being modelled require an

(A) HALE: Northrop Grumman Global Hawk (follow link to image source).

(B) MALE: Israel Aerospace Industries Heron. (follow link to image source).

(C) Tactical UAV: Thales Watchkeeper (follow link to image source).

(D) Light UAV: University of Southampton Spotter Light UAS (follow link to image source).

(E) Small UAV: Southampton University Laser Sintered Aircraft (SULSA) (follow link to image source).

(F) Micro UAV: FLIR Black Hornet PRS (follow link to image source).

FIGURE 2.2: Examples of the UAV categorisation provided in Section 2.2.2.

endurance and range that are generally closely matched fixed-wing UAS performances. However, there are rotary-wing aircraft that are capable of meeting these performance levels, for example the Schiebel CAMCOPTER S-100[3] which is currently being trialled in civilian services similar to those studied in the case study of this thesis.

[3]For more information on the Schiebel CAMCOPTER S-100 please see `https://schiebel.net/products/camcopter-s-100/`

2.2.3 UAS cost engineering

When considering the design of a UAS service, a critical component in the decision making is the cost of the UAS. Cost engineering is described by Humpherys and Wellman [28] as '*the application of scientific and engineering principles and techniques to problems of cost estimation, cost control, business planning and management science*'. Life-cycle costing of a product has been categorised by Asiedu and Gu [29] to include: research and development cost, production and construction cost, operations and maintenance cost, and retirement and disposal cost. In the case of a commercial off the shelf (COTS) UAV being costed for a service, the latter two categories are the most important (along with acquisition cost). Asiedu and Gu [29] go on to deconstruct these categories further, known as a cost breakdown structure (CBS), which is an important method for establishing the cost goals [30].

Operations and maintenance cost is a major contributor to the total life-cycle cost of the UAS. Maintenance is simply defined as '*ensuring that physical assets continue to do what their users want them to do*' by Moubray [31]. Papageorgiou [19] depicts a general maintenance programme to include:

- daily or routine inspections and preparation for flight;
- scheduled maintenance, based on a time interval (*e.g.* flight hours) and replacement of life-limited components;
- unscheduled maintenance generated by failures and findings during inspections.

All aspects of the maintenance programme generates costs. These costs can be either variable (based on the utilisation of the system), or periodic (based on routine inspections and the life-time of the components) [19]. This indicates that there is a relationship between maintenance cost and the Concept of Operations (CONOPS). Also, as the utilisation time increases, the maintenance cost is expected to rise as well. The failure mode associated with this is known as age-related failure and is often due to fatigue and corrosion. But most modern, complex systems have many different patterns of failure [31]. Therefore, the maintenance cost is also related to the reliability of the system.

Reliability is defined by Lewis [32] as '*the probability a system will perform its intended function for a specified period of time under a given set of conditions*'. From this definition it seems beneficial to create an optimal maintenance schedule based on the reliability of the system to reduce maintenance and inspection costs. This is known as reliability-centred maintenance (RCM) and is described by Moubray [31]. Several maintenance policies exist, for example:

- preventive - perform component overhauls or component replacement at a specific age limit regardless of its condition;

- condition-based - inspection of components periodically and replace them if they give identifiable indication that they are about to fail;
- run-to-failure - make no effort to anticipate or prevent failure and replace or repair when they do happen.

The problem of reliability optimisation and maintenance policy choice has been the subject of several studies, for example Bartholomew-Biggs *et al.* [33] studied the modelling of sequential imperfect preventive maintenance using age-reduction models[4] [34] to optimise the maintenance schedule. This demonstrates that not only the choice of policy can have a significant effect on the cost of a service, but also the schedule of maintenance. However, it is beyond the scope of this study to optimise the maintenance policy and its schedule.

To model the maintenance cost of a UAS, the system needs to be broken down into components and their reliability-behaviour translated to a suitable probability distribution. From this and a model of the chosen maintenance policy, a Monte Carlo simulation (MCS) can be performed to model failure times and maintenance times for individual components and the entire system. This can then be translated into a life-cycle maintenance cost and the uncertainty involved [19]. Both Schumann [35] and Papageorgiou [19] apply this technique using Weibull probability distributions of component-reliability when modelling the life-cycle maintenance cost of a UAS.

2.2.4 UAS modelling

To model the UAS to meet the needs of the optimisation challenge described at the end of Section 2.1, several performance metrics are required such as aerodynamic characteristics, engine characteristics and reliability characteristics. Depending on the level of detail required, the quantity of metrics vary. For low fidelity models the use of aerodynamic forces and moments to determine the acceleration of an aircraft is unnecessarily complicated and the source data may be difficult to obtain [36]. Duquette [36] offers a simple kinematics-based model of vehicle motion for UAV flight that requires minimal inputs[5] and can be tailored with additional inputs to determine acoustic signature and energy consumption rate. However, this model was designed for the purpose of testing the effectiveness of UAV command and control algorithms, not the energy consumption performance and reliability characteristics. The model is a time-driven simulation which gives continuous feedback on where the UAV is at any given time. This allows detailed

[4] Age-reduction models aim to fill the gap between a perfect repair and a minimal repair in the modelling of the stochastic behaviour of repairable systems [34].

[5] In the Duquette's paper [36] it states for forward flight it only requires airspeed, vertical speed, pitch angle and wind velocity. It goes on to state that, however, to include turning behaviour the addition of bank angle and headings is required.

data to be collected on performance throughout the mission with the knowledge of exactly when a value is reached (say, the level of fuel reaches 30%).

Goerzen [37] presented a motion planning algorithm which utilises the capability of the on-board sensors to guide a vehicle through an environment space to avoid obstacles and plan its own path. However, this method is unnecessary for the modelling of the UAV for the design of the UAS service. Moreover, the use of a time-driven simulation may not be the most efficient method (in terms of entire simulation run-time) for the entire scope of the problem (see Section 2.3 for more information on simulation methods).

Schumann [35] adopts an event-driven method for modelling the UAV. The simulation lines up an event, such as *move the UAV along a path to the next discrete point* and then it calculates the effect that event (the step from one point on the path the next) will have on the UAV based on, for example, linear interpolations of speed against power relationships or other performance models. If the UAV is capable of completing the event, it will go ahead and move on to the next event and so on. If it is not capable, then other actions are taken, such as refuelling. The inclusion of component-reliability using probability distributions can still be used as interrupt-events. However, even though this method does not provide continuous information, if the changes are constant between events and the same model has been used the result should not differ from a time-driven simulation. For example, if the energy consumption rate is constant, then the change in value between path segments will be the same as that calculated in the continuous version.

However, discrete-event simulation modelling is not well suited to a model of a payload with a continuous-sensor[6] can become an issue. If the sensor is required to find a target, for example during a SAR mission, ideally the user wants to know when the target is found rather than just which section of the path it was found on. One way to achieve this is to increase the number of sections of the path, but as this goes towards infinite points the discrete simulation becomes a continuous simulation and thus loses the advantages of it being discrete events. Alternatively, as the position of the target is known by the simulation, a calculation can be made to determine if the target will be detected by the sensor during that section (this is similar to the above method of determining if the UAV can complete the event). Then an interrupt-event can be scheduled for the target acquisition.

For event-driven modelling the performance of the UAS and payload need to be modelled to some effect. The following section delves into some of the possible methods, focusing on modelling the propulsion performance.

[6]Here, a *continuous-sensor* is used to describe one that is constantly acquiring data, such as an electro-optical (EO) sensor providing a video feed.

2.2.4.1 UAS propulsion performance

A generic propulsion performance model adopted by Schumann [35] linked energy consumption with speed. Then using the velocity as an input and linear interpolation between data points, the energy consumption value could be obtained. For example, Figure 2.3 displays the speed versus power of a small electric powered UAV (reproduced from a study by Ostler *et al.* [38]). From this a simple conversion to the desired unit can be performed, such as obtaining the mass of fuel used (by using its calorific value).

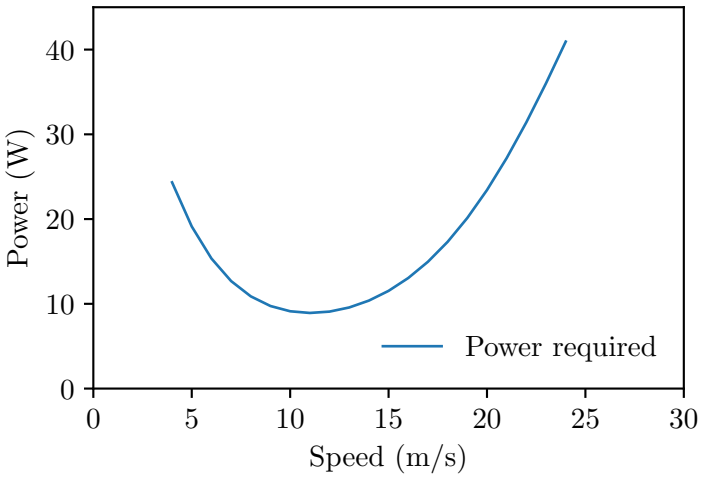


FIGURE 2.3: Example speed versus power graph of an electric powered small UAV. Graph reproduced from a study by Ostler *et al.* [38].

Schumann [35] lists the availability of data on various vehicle's energy consumption as an advantage of this method. However, he states there are many assumptions and simplifications made for this approach. Firstly, speed is not the only factor energy consumption depends on: drag, temperature, wind and other environmental factors also play a role in the energy consumption. Secondly, in the case of hydrocarbon fuel based UAVs, as fuel is used up and causes a reduction in the vehicle's weight[7] the energy consumption rate changes and should be taken account of using the Breguet range equation [39].

Internal combustion engine powered

To take this into account for an internal combustion engine (ICE) powered UAS the Breguet range equation for propeller driven aircraft, see Equation (2.1), can be applied [40] where R is the range, η is the propeller efficiency, SFC is the specific fuel consumption of the engine (the rate of fuel consumption divided by the power produced). This is usually given with the units g kW^{-1} h^{-1}, therefore requiring a conversion factor to match

[7]The reduction of weight as fuel is used is mostly applicable to vehicles that consume hydrocarbon fuel oils (such as avgas, petrol and diesel) where the change in weight compared to total vehicle weight is significant. On the other hand, as batteries are discharged, the change in weight is negligible compared to the total vehicle weight.

the standard units in the equation), $\frac{L}{D}$ is the lift to drag ratio and W is the weight of the aircraft:

$$R = \underbrace{\frac{\eta}{SFC}}_{\text{propulsion}} \overbrace{\frac{L}{D}}^{\text{aerodynamics}} \underbrace{\ln\left(\frac{W_i}{W_f}\right)}_{\text{structural}}. \tag{2.1}$$

The equation can be split into three main components: $\frac{\eta}{SFC}$ represents the effective propulsion of the aircraft; $\frac{L}{D}$ represents the flight regime and aerodynamics of the aircraft; and finally $\frac{W_i}{W_f}$ represents the structural properties of the aircraft (subscript i and f indicate initial and final respectively).

Commonly this equation is used to calculate maximum range by keeping the lift to drag ratio constant while flying at a constant speed. One way to achieve this is to make the cruise altitude steadily increase (hence air density will decrease) as fuel is consumed to keep $L = W$ and is known as the Breguet cruise-climb. However, in the case of UAVs the more common operational procedure is to maintain a constant altitude and constant speed due to the requirements of sensors and flight planning. Peckham [41] derives the range performance for these conditions using a drag polar of the form

$$C_D = C_{D_0} + kC_L^2 \tag{2.2}$$

where C_D is the drag coefficient, C_{D_0} is the zero-lift drag coefficient, C_L is the lift coefficient and k is the lift-induced drag factor. By maintaining constant altitude and constant speed in the range equation becomes

$$R = \frac{\eta}{SFC}\left(\frac{1}{\sqrt{C_{D_0}k}}\right)\left[\tan^{-1}\left(\frac{W_i}{qS\sqrt{C_{D_0}k}}\right) - \tan^{-1}\left(\frac{W_f}{qS\sqrt{C_{D_0}k}}\right)\right] \tag{2.3}$$

where S is the wing area and q is the dynamic pressure such that

$$q = \frac{1}{2}\rho U^2, \tag{2.4}$$

where ρ is the air density and U is the speed. From these equations (Equation (2.3) and Equation (2.4)) it is possible to see that the range, R, is a function of the following parameters

$$R\left(\underbrace{\eta, SFC}_{\substack{\text{engine}\\\text{performance}}}, \overbrace{U, \rho}^{\substack{\text{flight}\\\text{regime}}}, \underbrace{C_{D_0}, k, S}_{\text{aerodynamics}}, \overbrace{W_i, W_f}^{\text{structural}}\right) \tag{2.5}$$

By rearranging Equation 2.3 and assuming SFC and the propulsive efficiency, η, are constant it is possible to find the weight of fuel required for the UAV to fly a given distance at constant speed U and constant altitude corresponding to the air density ρ. However, Peckham [41] and Ferraro [42] note that there is an error in the assumption of constant SFC and η because when the aircraft reduces in weight due to fuel burn, the throttle setting must be adjusted to maintain constant speed and altitude.

Battery powered

In the case of battery powered UAVs the weight of the vehicle does not reduce as battery power is converted (see Footnote 7 on page 18). Therefore, a simpler equation can used to predict the range and endurance. Methods have been put forward in several ways, for example by Retana and Rodríguez-Cortés [43], but as pointed out by Traub [44], they often are not presented in a manner consistent with the normal methods adopted by the aeronautical community. In order to do this, Traub introduces an equation that is based on the power delivered by the propulsion system being equal to the power required to overcome the drag [44]. Traub also accounts for the Peukert effect [45] on battery capacity [46]. This is where the effective capacity of the battery is dependant on the current draw. The significance of this effect is discussed in [44] and is further validated in [47].

To implement the equation introduced by Traub, using the same drag polar presented in Equation (2.2), the power required P_{req}, assuming steady level flight, can be stated as

$$P_{req} = \frac{1}{2}\rho U^3 C_{D_0} + \frac{2W^2 k}{\rho U S}. \tag{2.6}$$

Then, using a modified Peukert equation [44] which accounts for the effect of discharge rate in the form of

$$t = \frac{Rt}{i^n}\left(\frac{C}{Rt}\right)^n \tag{2.7}$$

where t is the time in hours, i is the discharge current in amperes, C is the battery capacity in ampere-hours, n is the discharge parameter dependent on battery chemistry and temperature and Rt is the battery hour rating in hours.

This can be substituted into the equation for battery output power P_B such that

$$P_B = V\frac{C}{Rt}\left(\frac{Rt}{t}\right)^{1/n} \tag{2.8}$$

where V is the battery voltage. From here the two power equations (Equation (2.6) and Equation (2.8)) can be equated while taking into account the efficiency of the propulsion system as η_{tot} and this can be solved for t resulting in

$$t = Rt^{1-n}\left[\frac{\eta_{tot}VC}{P_{req}}\right]^n \tag{2.9}$$

where t represents the endurance in hours. Finally, the range can be found by multiplying Equation (2.9) by the speed, U.

However, this method does not account for the drop in voltage as the battery is discharged and during a validation experiment the predicted endurance was found to be 10 - 14% higher than the experiment when the initial voltage was used. When the average voltage during discharge was used the error was reduced to approximately 3% [47]. Therefore, if only the initial voltage is known then a reduction of $\sim 12\%$ should be used for predicting the endurance and range (this reduction method was demonstrated in [48] to good effect).

Finally, the battery discharge parameter, n, needs to be known or assumed. This can vary between models and manufacturers. Lithium-polymer batteries typically have a discharge parameter $n = 1.3$. Although in [49] it is suggested that lithium-ion batteries act closer to an ideal battery (therefore, $n = 1$) and that the Peukert equation is only applicable when the battery is discharged at a constant temperature and constant discharge current otherwise the result is an underestimation of the remaining capacity[8].

2.2.5 UAS payload performance

The payload which is defined by Austin [18] as '*only that part of an aircraft which is specifically carried to achieve the mission*' is also an important component to model: its performance influences the outcomes of the mission. Therefore, it is important to model the payload's capabilities as best as possible.

Firstly, it is worth considering the range of payload types. The most common UAS payload types are EO sensors for ISR, aerial filming, SAR missions. However, other payloads applications include atmosphere/pollution monitoring systems, radio-relay systems, public-address systems, disposable payloads (such as crop-spraying, fire-fighting, releasing research equipment [50] or humanitarian-aid drop systems [51]) and cargo/transportation [18, 35]. All the aforementioned payloads can be defined as either *active* or *inactive*

[8]Doerffel [49] suggests that the capacity obtained from a lithium-ion battery is strongly dependent on temperature which, in turn, is dependent on the rate of discharge and therefore it is this factor that can increase the available capacity. Nevertheless, the validity of this is questioned by Traub [44].

where inactive corresponds to those that only being transported (for example cargo), and active refers to those that gather data or have an effective area, volume or field of view (FOV)[9].

Austin [18] states that there are several factors which can drive the design of an active payload (focusing solely on EO sensors) such as

- the range, endurance and altitude of the platform;
- the range and area of surveillance needed;
- the resolution of the imagery;
- the need for tracking;
- the need for on-screen display (OSD) of latitude/longitude, date and time data.

Moreover, these design factors can often become requested features by the clients and stakeholders creating the RFP.

When considering EO sensors Schumann [35], Ferraro [42] and Surendra [52] all base their modelling on the works from Leachtenauer & Driggers [53] and Gundlach [54]. The EO system is a collection of individual detector elements in an array, known as the focal plane array (FPA) with H_{pix} horizontal detector elements and V_{pix} vertical detector elements. The FOV in a given direction is defined as

$$FOV = 2\tan^{-1}\left(\frac{d}{f}\right) \tag{2.10}$$

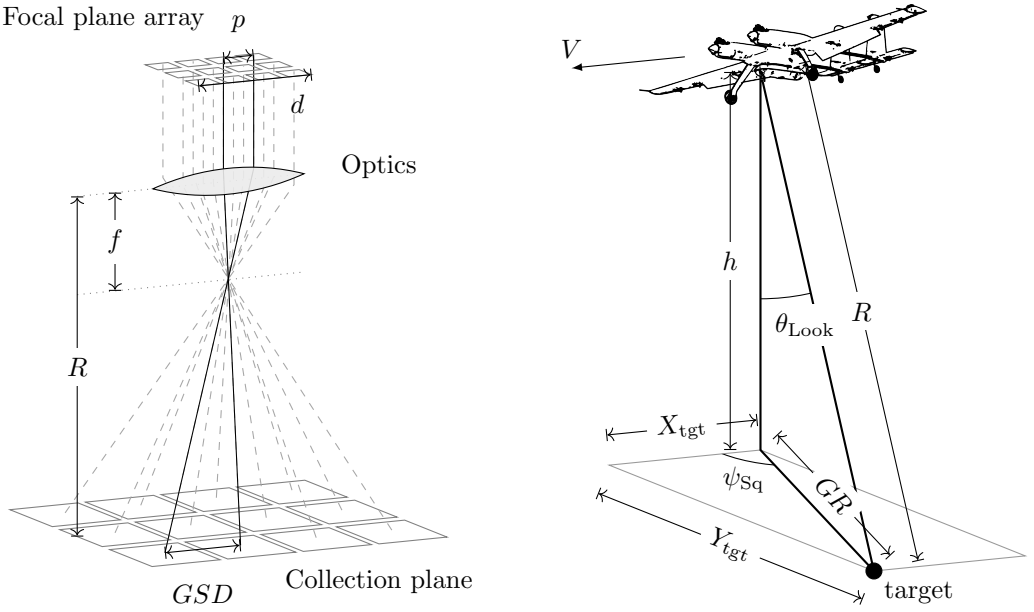
where d is the length of the focal plane array and f is the focal length (see Figure 2.4a).

The image resolution is governed by the ground sample distance (GSD) parameter which is a function of the camera FPA, optics, and collection geometry (see Figure 2.4a). GSD is the distance between pixels projected on the ground (collection plane) at slant-range R which can be obtained as

$$R = \sqrt{h^2 + GR^2} \tag{2.11}$$

where h is the UAV altitude and GR is the ground range from the target to the UAV (refer to Figure 2.4b for geometries). As the FPA is usually rectangular the GSD for

[9]To clarify, EO sensors and pollution monitoring systems gather data and have a specific FOV or volume respectively in which they are effective (*i.e.* contribute to the success of the mission). However, for example, crop-spraying does not gather data, but once spraying is commenced it has an effective area on the ground. Therefore, it can be considered active.



(A) Basic optics geometry.

(B) Basic geometry for remote sensing.

FIGURE 2.4: Diagrams defining parameters regarding optical payload modelling. Both diagrams reproduced from Gundlach [54].

horizontal and vertical can be calculated separately with the horizontal GSD at the centre of the image, GSD_H, as

$$GSD_H = \frac{p}{f} R \tag{2.12}$$

where p is the detector element pitch, the distance between the detector elements' centres (sometimes known as the pixel pitch), assuming the horizontal row in the FPA is aligned with the horizon. Similarly, the vertical GSD at the centre of the image, GSD_V, is

$$GSD_V = \frac{p}{f \cos(\theta_{\text{Look}})} R \tag{2.13}$$

where θ_{Look} is the look angle. By rearranging Equation (2.10) and substituting into Equations (2.12) and (2.13) we get

$$GSD_H = 2 \tan\left(\frac{FOV_H}{2\ H_{\text{pix}}}\right) R \tag{2.14a}$$

$$GSD_V = \frac{2 \tan\left(\frac{FOV_V}{2\ V_{\text{pix}}}\right)}{\cos(\theta_{\text{Look}})} R. \tag{2.14b}$$

This gives a measure of resolution of the imagery. However, to predict the quality and utility of the imagery an image quality metric has to be introduced. In previous works

[35, 42, 52] the Johnson criteria was adopted to predict the probability the object of interest (labelled *target* in Figure 2.4b) will be detected, recognised and identified based on the sensor's resolution. The Johnson criteria creates a characteristic dimension, d_c, for the target such that

$$d_c = \sqrt{W_{\text{tgt}}\ H_{\text{tgt}}} \tag{2.15}$$

where W_{tgt} is the target's width and $H_{\text{tgt}}$is the target's height (assuming the target is seen by the UAV as a bi-dimensional object). Using this and the sensors average GSD, the number of cycles, N, across the target can be calculated as

$$N = \frac{d_c}{2\ GSD_{avg}}. \tag{2.16}$$

The number of cycles of a target is derived by substituting the target image with pairs of black and white lines along the target's characteristic dimension. Each cycle is two pixels (one for black, one for white) where a pixel is the GSD, hence can be treated as $2\ GSD_{avg}$. Using this idea of cycles, the probability of achieving the discrimination task for a given number of cycles is given by the empirically found equation

$$P(N) = \frac{\left(\frac{N}{N_{50}}\right)^{2.7+0.7\times(N/N_{50})}}{1+\left(\frac{N}{N_{50}}\right)^{2.7+0.7\times(N/N_{50})}} \tag{2.17}$$

where N_{50} is the number of cycles that corresponds to a 50% discrimination probability [54]. N_{50} is given as 0.75 for detection, 3.0 for recognition and 6.0 for identification[10].

However, it is common to get a sensory requirement in the RFP given in terms of a NIIRS (national imagery interpretability rating scale) performance value [55] - 'a series of government standard qualitative metrics that help characterise the intelligence value of an optical system under collection conditions' [54]. Therefore, this image-quality metric will be considered here too.

Irvine [55] present the full set of scales and ratings in his paper. However, the scale is subjective and may vary among analysts, for example an attribute of NIIRS rating 7 is 'Identify fitments and fairings on a fighter-sized aircraft' [55]. Therefore, the general image quality equation (GIQE) was developed as a tool to provide NIIRS performance predictions for new equipment still under design [56]. The GIQE is expressed as

[10]Detection is define in Gundlach [54] as a reasonable probability that an imagery feature is of a general group (*i.e.* an aircraft); recognition as a class of the group (*i.e.* a fighter aircraft); and identification as object discrimination (*i.e.* Tornado GR4)

$$NIIRS = c_0 + c_1 \ \log_{10}(GSD) + c_2 \ \log_{10}(RER) + c_3 \ \frac{G}{SNR} + c_4 \ H \tag{2.18}$$

where

$c_0 - c_4 =$ constants defined in Table 2.1,
$GSD =$ geometric mean ground sample distance (in inches),
$RER =$ geometric mean normalised relative edge response,
$G =$ post-processing noise gain,
$SNR =$ signal-to-noise ratio of the unprocessed imagery,
$H =$ geometric mean system post-processing edge overshoot factor.

TABLE 2.1: Table of NIIRS constants. Reproduced from Gundlach [54].

	c_0	c_1	c_2	c_3	c_4
$RER \geq 0.9$	10.251 (visible)	-3.32	1.559	-0.334	-0.656
$RER < 0.9$		-3.16	2.817		

The issue with the GIQE is that several of the parameters require detailed knowledge of the imagery system performance and the collection environment [54]. To aid this, Leachtenauer *et al.* [56] provides a table of the range of values in the overall NIIRS data set that was used to develop the GIQE and Gundlach comments on the parameters that have a small impact on the NIIRS value, such as the H range given in [56] gave a range of 0.321 NIIRS.

2.2.6 UAS modelling summary

The sections above have presented ways that the cost and performance of a UAS can be modelled. All that has been presented is relevant to the work in this thesis. The model and simulation created to assist in the design of a UAS service should take all these areas into account at the appropriate level of detail and fidelity for the model based on the available inputs.

The performance of the payload was also presented above. The main focus was on the use of EO sensors due to their common use in UAS applications. The NIIRS rating was discussed due to its prevalence in RFPs. However, the ability to apply this rating to a model can be a challenge if the sensors details are limited. Therefore, the Johnson criteria should be used as the basic model to enable the payload sensor model to mimic the detection of targets.

In Chapter 5, the UAS service simulation tool developed using the UAS modelling techniques discussed above is demonstrated via a case study. In the case study, the

UAS service modelled represents a maritime search service where the UAS flies from its operating base to a mission location, performs a task, and then returns to its operating base. By implementing the propulsion performance models described in Section 2.2.4 it also allows for specific constraints to be included, such as weather conditions (*e.g.* wind speed) and energy limitations (*e.g.* fueling policies). The effect of these constraints then feed into the operational and maintenance costs in the life-cycling costing model presented in Section 2.2.3.

2.3 Overview of modelling and simulation

This section provides an overview of modelling techniques, concepts and categorisation. Modelling was described by Maria [57] as '*the process of producing a model; a model is a representation of the construction and working of some system of interest*'. These systems of interest could be '*ideas, objects, events, systems or processes*' according to Gilbert *et al.* [58]. Law and Kelton [59] provide an objective overview of the ways to study a system as shown in Figure 2.5. This overview highlights that models can be either a physical model or a mathematical model with either an analytical solution or one that requires simulation.

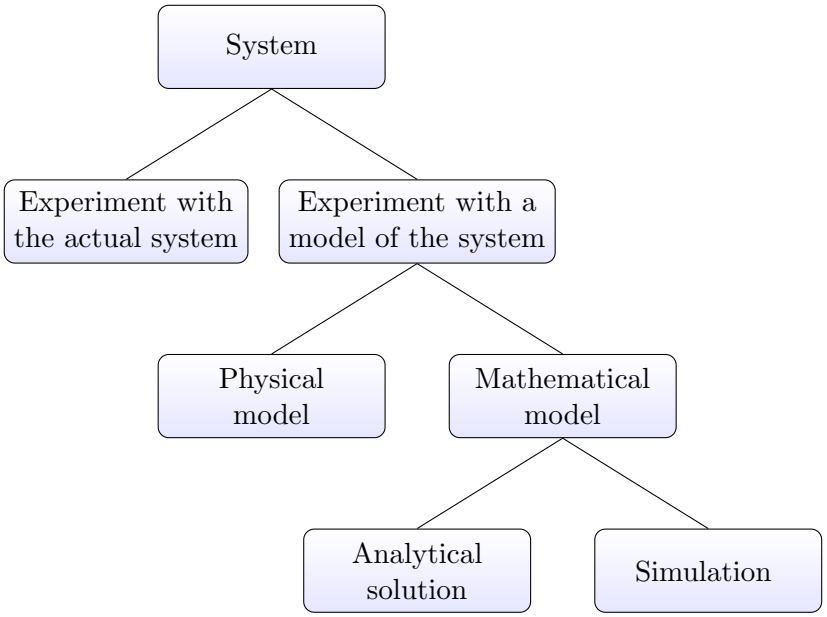


FIGURE 2.5: System modelling overview. Reproduced from Law and Kelton [59].

The use of modelling has become common practice in many research disciplines today; management and social sciences, economics, engineering, chemistry, biology, physics, medicine and healthcare to name a few. For example, with the world becoming more interconnected, models are required for international commercial operations to run efficiently, for targeted advertising to be effective and for financial risk to be understood.

Computational modelling has allowed the study of problems where actual experimentation is impractical, inefficient or has a disproportionately high cost, and where the problem can not be solved directly [60]. Many of these problems are complex systems where modelling is often the only tool available for understanding how they work [7]. To develop models of these complex systems the selection of the correct modelling paradigm, its development, integration and implementation need to be carefully thought through.

2.3.1 Simulation categorisation

Simulation, or '*numerically exercising the model*' as put by Law and Kelton [59], has many different methods which can be categorised in several ways. Rubinstein [61] and Law & Kelton [59] offer up the same three classifications, two of which are: *static versus dynamic* and *deterministic versus stochastic*. However, the third and most common categorisation amongst journal papers and textbooks is the *representation of time and state in the simulation model* as put forward by Nance and Sargent [62], or described as *continuous versus discrete* by Rubinstein [61] and Law & Kelton [59]. This classification can be seen in Shannon's [63] 1977 layout of simulation techniques, adapted in Figure 2.6, which was originally used as a diagram for assigning programming languages to simulation techniques.

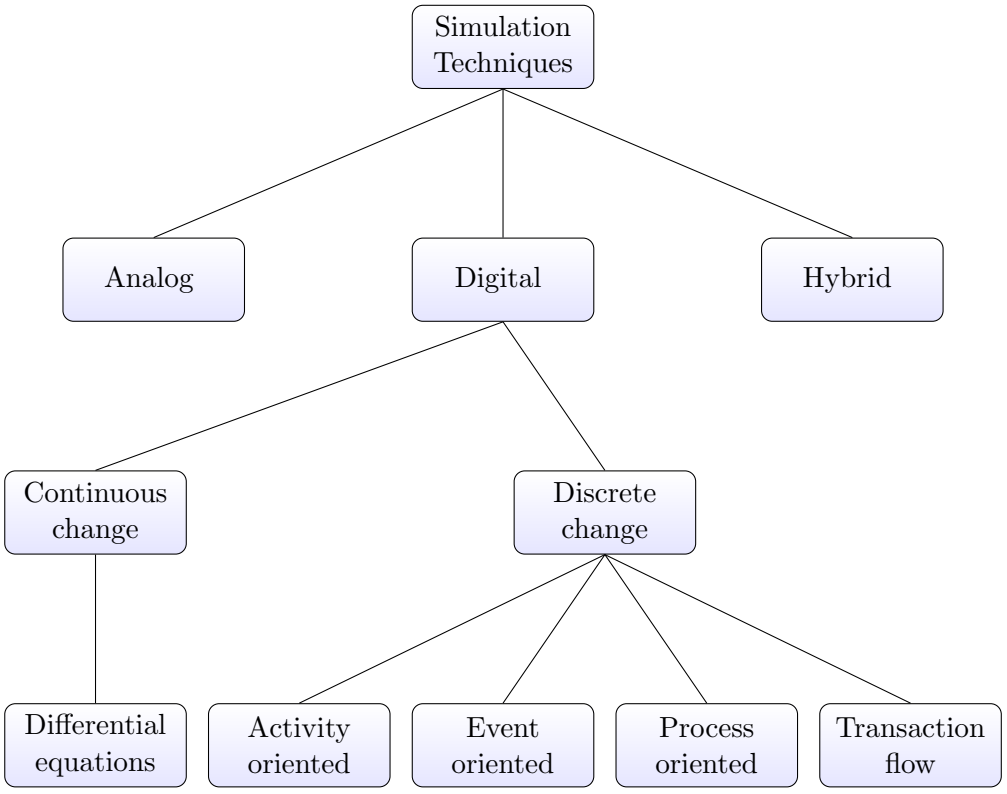


FIGURE 2.6: Simulation techniques. Adapted from Shannon [63].

Adapted versions of Shannon's diagram focused on digital techniques, combined with the categorised modelling paradigms from Borshchev and Filippov [64], can be seen in Yu [7],

Jinks [65] and Schumann [35] with Jinks' adaptation shown in Figure 2.7. This version clearly lays out the two classifications, 'time-driven' and 'event-driven' (which correspond to 'continuous change' and 'discrete change' in Shannon's diagram respectively), under which the four modelling approaches are headed: system dynamics, continuous[11], agent based and discrete event.

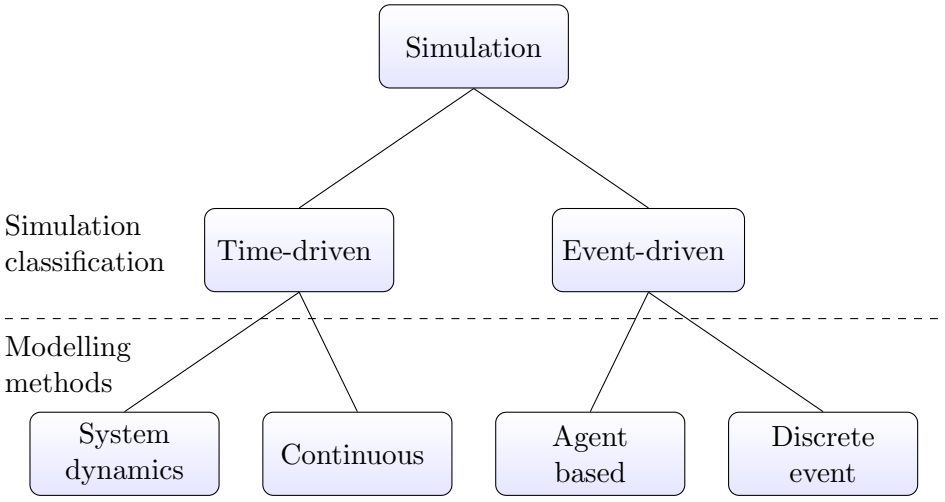


FIGURE 2.7: Simulation classification and modelling methods. Reproduced from Jinks [65], an adaptation from Yu [7].

2.3.2 Time-driven methods

Time-driven modelling methods are where the time component of the simulation is advanced at fixed, regular intervals and the state variables are recalculated at each time step giving the appearance of continuous change. On a technical point, due to the nature of digital computing, continuous changes are not physically possible as the time component has to be discretised. However, the size of the discrete interval will be set such that in a simulation it would be considered continuous [7].

As mentioned in Section 2.3.1 and Footnote 11, continuous modelling involves differential equations of the state variables over time, such that their values can be predicted with certainty [64, 7]. This method is commonly used when a high level of output detail is required, such as the finding the velocity and position of a bouncing ball.

System dynamics, on the other hand, is where real-word processes are characterised in terms of stocks, flows and feedback loops. This method, developed by Jay Forrester [66], deals with high levels of abstraction to model, for example, the dynamics of a global population where the causal relationships are presented using differential equations to form interacting feedback loops.

[11] In Borshchev and Filippov the 'continuous' approach is entitled 'dynamic systems' and is described as 'used to model and design "physical" systems'. This means that the state variables of the dynamic system relate to a direct physical meaning: location, velocity, pressure, *etc.* and hence are inherently continuous [64].

2.3.3 Event-driven methods

Event-driven modelling methods are where the simulation advances from event to event in variable time steps such that the state variables change instantaneously at separate points in time [59]. The event is defined by Law and Kelton [59] as 'an instantaneous occurrence that may change the state of the system'. They go on to say that the word *may* is used because the event might not actually result in a change in the state of the system, for example, the event might trigger the end of a simulation run at a set time.

2.3.3.1 Discrete event modelling

Discrete event modelling (DEM) is based on the concept of entities, resources, constraints and block-charts describing entity flow and resource sharing [64]. It can be seen as logical sequences of possible activities, where the entities can represent parts, people, messages, *etc.* These activities are often abstracted as time delays, such as a customer talking to a shop assistant, thus using a resource (the shop assistant) and possibly creating a queue. The block-charts consist of activities such as queues, delays, processes, seizing of resources, releasing resources, *etc.* Between events, the underlying assumption of discrete event modelling is that nothing of consequence occurs [7]. Because of this DEM is often used for the study of manufacturing plant operations (Sajadi *et al.* [67]) and supply chain design (Chen *et al.* [68]).

DEM is suited to models with a medium level of abstraction [64] where the process can be described as top-down flowcharts. The fundamental concepts behind DEM are the simulation objects - which include the entities and the activities (such as queues and timers); and the event - which acts on the simulation object (such as changing its state or scheduling future events) [7]. In complex models where the number of events is large, the processing, storing, sorting and accessing of these events becomes significant in terms of computational time. Therefore, careful selection of how the event list is handled is required.

One solution to decrease the computational time to run a large discrete event simulation (DES) is to paralellise the program (*i.e.* run a single simulation program across multiple processors). However, this is challenging due to the precedence constraints (that dictate which event must be performed before which others) are quite complex and highly data dependent. This is also known as the synchronisation problem [69]. This is well explained by Fujimoto [70], and in [69] some of the recent research into parallel and distributed simulation programs are reviewed: the two main methods used are conservative algorithms and optimistic algorithms.

2.3.3.2 Agent based modelling

Agent based modelling (ABM) is where the control of the model is decentralised and distributed among the agents. Rather than define the global system behaviour, the modeller defines how the individuals behave using rules to control reactions to environmental inputs, self-learning and communication [64]. This creates a level autonomy which can lead to unpredictable emergent behaviour which may be of interest in some areas of research such as the study of a social science problem. Yu [7] notes that unpredictable emergent behaviour may not be of benefit to engineering problems as repeatability and predictability are wanted characteristics from a simulation.

There appears to be no fully agreed upon description of an agent across the disciplines [64]. However, the one offered by Jennings [71] based on Wooldridge's [72] description is concise: '*an agent is an encapsulated computer system that is situated in some environment and is capable of flexible, autonomous action in that environment in order to meet its design objectives*' [71]. The use of the term *autonomous* is one of the key characteristics that distinguishes an agent from an object. An object will respond predictably and will be completely obedient, whereas an agent (which has control over its internal state and its own actions) will make a choice on its action following an event that best suits its objectives [71]. A more recent paper by Macal [73] offers up four informal definitions based on applications appear in literature in a bid to distinguish ABM from other modelling and analytical approaches. These step though the features and requirements generally acknowledged as components of an ABM (individual, autonomous, interactive and adaptive).

For a deeper understanding of ABM, both Macal [73] and Wooldridge and Jennings [74] provide overviews of the important theoretical and practical issues associated with ABM. In Wooldridge and Jennings' overview, agent architectures are discussed and sorted into two main categories: deliberative and reactive (with the possibility of hybrid architecture too).

Deliberative architectures (also described as *classical approaches* by Wooldridge and Jennings [74]) contains agents that behave more like they are thinking and there is a consideration of alternative courses of action before an action is taken [75]. This is a common technique for social science modelling due to its ability to effectively model human behaviour [76]. Whereas in a reactive architecture (highlighted as an *alternative approach* in Wooldridge and Jennings [74]) the agents respond to events that occur in the environment without engaging in complex reasoning [74]. This approach was used by Schumann as it allowed intuitive model building for real systems with limited system knowledge and agents within the work only follow operational rules [35].

2.3.4 The vehicle routing problems

The family of vehicle routing problems (VRPs) is defined succinctly in Irnich, Toth and Vigo [77] (in terms of the generic problem) as: given a set of *transportation requests* and a *fleet of vehicles*, the task is to determine a set of vehicle routes to perform all (or some) transportation requests with the given vehicle fleet at minimum cost; in particular decide which vehicle handles which requests in which sequence so that all vehicle routes can be feasibly executed [77].

This set of problems is well researched, partially due to the notorious difficulty in solving combinatorial optimization problems, and partially due to their increasing appearance in real-world applications (*e.g.* providing a next-day delivery service). The VRP was introduced in 1959 by Dantzig and Ramser [78] as a practical application and since then many groups have research this field, presenting exact or heuristic (and meta-heuristic) solutions to the multiple variants of the problem.

A basic variant of the VRP is the *capacitated* vehicle routing problem (CVRP) in which the transportation requests consist of the delivery of goods from a single depot to a given set of locations and then back to the depot, and each location has a given amount of goods to be delivered (*e.g.* the weight of the goods). The fleet in this basic variant is assumed to be homogeneous and therefore all the vehicles have the same capacity. An example of this would be a fleet of equivalent vans from one warehouse supplying medical supplies to pharmacies of different sizes. The problem statement of this variant is mathematically defined in [77] and the important mathematical programming formulations are presented.

Several variants and constraints can be applied to this basic variant to tailor the problem to real-world applications. In the case of this thesis, the most relevant variant of the VRP is the Heterogeneous or mixed Fleet VRP (HFVRP) - see Baldacci *et al.* [79] for a survey of the research and work on this variant. In the HFVRP each vehicle in the fleet has a potentially different capacity and cost. This variant combined with the additional constraint of multiple depots is a generic description of the layout of UAS assets, mission locations and operating bases in case study presented in Chapter 5. Irnich, Schneider and Vigo [80] breaks the HFVRP into two strains, one focused on the strategic issue of finding the best assortment of vehicles to be used for the long term sizing of the fleet (often referred as Fleet Size and Mix (FSM) problems), and the other focused on the tactical issue of using the most appropriate vehicle from a limited fleet (referred as Heterogeneous VRP (HVRP)).

However, one of the major issues of applying a VRP model to a UAS service model is that the majority of VRP models require prior knowledge of the locations of the transport requests (mission locations in the context of this thesis) to build up the routing and provide a solution to the problem. The main type of UAS service being modelled in this thesis is that of dynamically generated transport requests (both in location and frequency).

For example, Zipline's response to a request for medical delivery discussed in Section 1.1. There is research towards dynamic VRP (DVRP) and Ojeda Rios *et al.* [81] provides a survey of the literature produced between 2015 and 2021 focusing on applications and solution methods. One interesting metric produced by Ojeda Rios *et al.* when categorising the articles related to applications was that 17.5% were related to studying services. This could be worth exploring as an alternative method to tackle the problem. Some issues of non time-based models can be overcome by existing non-dynamic variants of the VRP by tailoring them to the real-world application under consideration. For example, missions overlapping in time can be included through time-window constraints to force multiple vehicles to become utilised. This type of UAS service often has multiple depots (*i.e.* operating bases in the case study presented in Chapter 5), and very few transport requests active at any one time (*i.e.* missions in the case study presented in Chapter 5). Therefore, the routing element of the VRP in this type of UAS service is not the main difficulty. Instead, the challenge is combining the strategic issue of fleet size and mix with the tactical issue of using the most appropriate vehicle from a limited fleet.

VRPs, their variants and associated models could be applied within the UAS service model as a means to determine the daily tasking of the UAS if the service required asset routing. However, this type of UAS service was not the focus of this thesis, but could be included in the mission-generation architecture. This would allow a greater variation in the type of UAS services that could be studied using the decision tool being developed here.

2.3.5 Comments on simulations

Using a simulation has both advantages and disadvantages. This section aims to provide a few comments on the use of simulations. Firstly, it should be noted that despite the obvious cost advantage of simulation over physical modelling, simulation is time-consuming in design, implementation and analysis, and often requires experts throughout the whole process [57, 61]. Moreover, both time and experts come at a cost. This needs to be accounted for when considering the need for a model and simulation, and setting the level of complexity and scope of the model.

Several textbooks and journals advise that the simulation model should be kept as simple as possible: this allows them to be more understandable - in both model structure and result interpretation, faster to develop, more flexible and faster to run [82, 83, 84]. Simple is defined in a handful of ways by Innis and Rexstad [83]: shorter, more transparent and more efficient. Following this, several journal papers have attempted to show or report on the relationship between model confidence (or effectiveness) and the level of details (or articulation) [85, 86, 87, 84]. The illustrative diagram in Figure 2.8 reproduced from Lobão and Porto [86] demonstrates the relationship. The general consensus is that there is an optimum effectiveness for models where the model is not too simple nor too detailed.

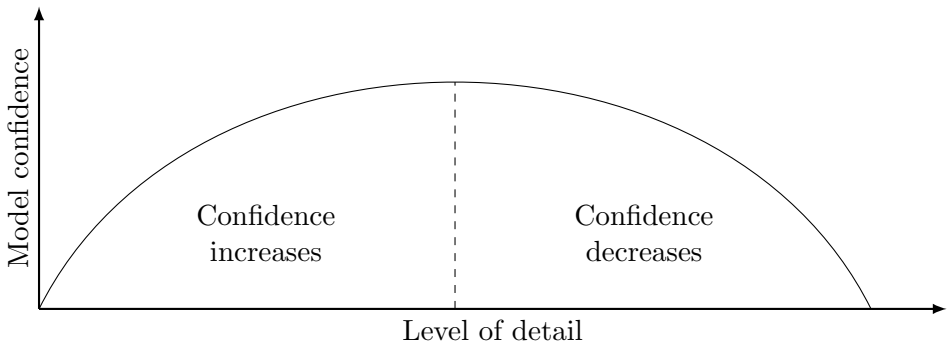


FIGURE 2.8: Relationship between level of detail and model confidence. Reproduced from Chwif *et al.* [84], original from Lobão and Porto [86].

A simulation technique, which opposes some of the gains simplicity offers, is the use of an animated simulation. Amiry [88] is identified by several papers and books as being amongst the first to add animation to his simulation of information flow in a steel-making plant to aid the user (in his 1965 paper). Hurrion [89] published his PhD thesis on the potential for simulations that are both visual and interactive. This has now become a common method in most modelling software (Extend, AnyLogic, *etc.*), allowing the user to visualise the workings of the model and to interact with it, for example select an entity and obtain more information on its status. Robinson [82] highlights some benefits of visual interactive simulations as: greater understanding of the model; easier model verification; improved understanding of the results; and more intuitive to non-simulation experts. Therefore, it not only benefits the developer with visualising modelling errors, but it also benefits those for whom the simulation was designed to inform [90]. However, as mentioned earlier, animation will increase the complexity of the model and make the simulation run slower. Although, once the developer is satisfied by the model and how it is working, the animation can be switched off which should improve the simulation run-time.

2.3.6 Modelling and simulation summary

Modelling can be achieved in a variety of ways (for example, physical, analytical or simulation models) each with their advantages and disadvantages. Often, simulation techniques are chosen due to the other options being impractical, inefficient (in both cost and time) or impossible. They also offer an insight into how a complex system works.

Simulations can be categorised based on the representation of time and state in the model, which leads to continuous and discrete models or time-driven and event-driven simulations. Both model paradigms have their advantages and disadvantages as presented above. The choice of modelling method should usually be based on the closest conceptual match and an inappropriate choice may result in inaccurate results and possible computer inefficiencies [7].

In the case of modelling a UAS service, the use of a discrete event simulation is preferable due to the system consisting of events and resources. It would be computationally inefficient to model this with a time-driven model. Therefore, an event-based model has been pursued in the work for this thesis. This has been combined with an agent-based model using a reactive agent architecture (*i.e.* no complex reasoning, purely following operational rules).

Care needs to be taken when formulating the model to ensure it contains the correct level of detail to ensure the model is understandable, easier to develop and more flexible to adjust. It should also be designed such that the results can be easily interpreted and the model is as transparent as possible - for both debugging and verification purposes. To achieve this, simple animations or visual representations are a useful tool. However, this introduces computational inefficiencies, therefore the simulation should be able to run with them switched off.

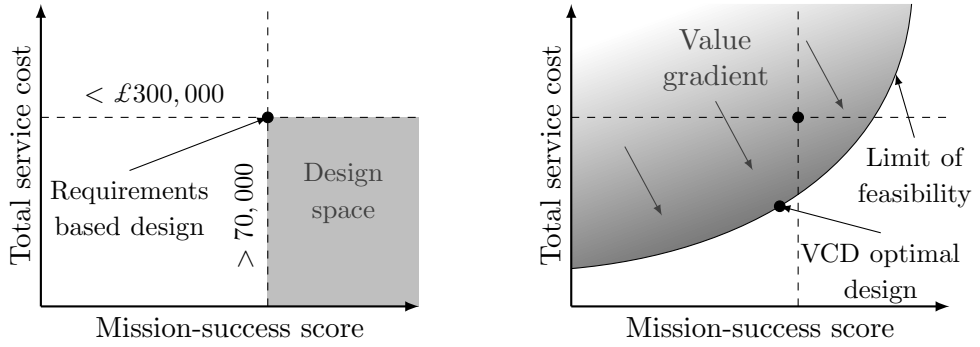
2.4 Overview of design decision methodologies

The design of a UAS service involves the comparison and ranking of design alternatives, each comprised of multiple attributes with corresponding objectives. Inevitably, improving one of these objectives may worsen another. The links and trade-offs between attributes are often not straightforward. Hence, the design process becomes a multi-attribute decision methodology (MADM) problem, where the attributes need to be well-defined, such that the objectives can be fully represented [52]. The development of system-value-models aimed to aid this process and guide the decision maker to reach a final solution in a systematic way by integrating all these performance and characteristics into a single figure of merit [42]. A comprehensive review of value-centric design methodologies (VCDMs) is provided by Ross *et al.* [91] and a survey of aerospace value models is given by Collopy [92].

This section will introduce the concept of value-centric design (VCD) and provide a overview of different MADMs and VCDMs that are available in the literature. It is worth pointing out that both MADM and VCDM are inherently the same, as both aim to find the best solution from a set of alternatives based on the system's attributes and preference structure, but with slight differences when comparing the choice of objective function as described in Surendra [52]. Also, it should be noted that some authors in the literature use different terminology for VCDMs (value-centric [93], value-driven [11] and value-based [94] methodologies) and assign slightly different meanings to each. Here, however, they will be used interchangeably.

2.4.1 Value centric design methodologies

Value centric design uses economic theory to improve the design of large systems through optimisation [11]. The main improvement of VCDMs in comparison to traditional systems engineering approach is due to its ability to systematically evaluate design alternatives in terms of cost and benefits [91, 42]. For example, a traditional systems engineering approach may be to optimise the system to achieve minimum cost while meeting a desired performance level (see Figure 2.9a). These high level requirements will cause certain high level parameters to become fixed in order to meet the requirements. Therefore, when it comes to designing the subsystems there is very little room for manoeuvre in the design space [95]. This means if a high level requirement is not met (or is changed) the program could require an extensive redesign which costs time and money. Real-world examples of this problem occurring are given in Collopy & Hollingsworth [11].



(A) Traditional systems engineering approach.

(B) Value-centric design approach.

FIGURE 2.9: Comparison of a traditional systems engineering approach where the design space is defined by requirements, against a value-centric design (VCD) approach where the design space is defined by values and technology. Figure adapted from Collopy [96] and Cheung *et al.* [97].

In comparison, a VCD approach would aim to capture the complex interactions between each sub-system as design changes are made. This is achieved by assigning a value to a given design as a score of 'goodness' with the aim to maximise this score [97]. The value is calculated through the system value model which involves the identification of the stakeholders' needs and the figures of merit that describes the design (*i.e.* an objective function). This value can then be used to compare and rank all the design alternatives possible within the design space [98] (see Figure 2.9b).

To illustrate this, see the example drawn in Figure 2.9. Here, the example has been tailored from [96, 97] to suit the problem discussed in this thesis. By plotting the design space of *total service cost* against a *score of mission-success* we can place every possible design. If we then, using the traditional requirements-based approach, state some requirements we form an area where our designs have to sit but with no indication on where in the area the design should be. Therefore, it can be seen that by using the traditional approach we can select a design that meets both requirements, see Figure 2.9a

. However, by using the value-based approach, we can assign a value to each design based on a value function and use this to compare designs. The resultant plot of design and value is shown in Figure 2.9b and provides an optimal design and a value gradient. The value gradient is produced by the partial derivative of the design value with respect to the components that formed the value function (*i.e.* service cost and mission-success score in this case). The optimal design falls marginally short of the requirement of mission-success score, but yet costs nearly half the price. The line indicated as the 'limit of feasibility' represents the Pareto front. This is where none of the attributes can be improved with out worsening at least another. The candidate designs that sit on this front are known as the Pareto set and this will contain the optimal design.

2.4.2 Implementation of value-centric design

There are several different methods to implement VCD but all follow the same general framework [97, 42, 19]:

1. Define the problem and identify the stakeholders.
2. Define what 'value' means to the stakeholders (and define stakeholders hierarchy if necessary [92]).
3. Define the system to be designed in terms of quantifiable attributes.
4. Create the value model to coherently measure the value of the design alternatives.
5. Generate the candidate systems through component models.
6. Measure the value of the candidates using the value model.
7. Perform design optimisation and trade-off studies to find the 'optimal' solution.

2.4.3 Stakeholders

The stakeholders of a UAS service are all those involved in the requisition, design, implementation, or interaction of the service[12]. Often, the stakeholders will be interested in different attributes of the system. Therefore, it is desirable to aggregate the preferences, however this has its problems as stated by Arrow's general impossibility theorem [99]. The use of game-theory to aggregate the preference of multiple stakeholders has been studied by Papageorgiou *et al.* [100]. Other methods have included taking the geometric mean of the groups preferences [101].

[12]This could include people in the environment in which the service will operate. For example, people under the approach path to the landing-site could have a preference in noise levels and safety during the higher risk manoeuvres.

2.4.4 Multi-attribute decision making

To build the objective function the desires and aversions of the stakeholders need to be established and ranked to provide a weighting to the attributes through the use of MADMs. The input data to these methods can be either qualitative or quantitative: for example, a judgement by the stakeholder, or an evaluation through a model respectively. The task of formulating a single, overall objective function of multiple-attributes is very challenging. One way to simplify the problem is to deconstruct the overall objective function into multiple single-attribute objective functions. Then the overall objective function can be reconstructed (additive, multiplicative, multilinear) depending on the assumptions made. Sen & Yang [102] provide a classification of different MADMs based on the acquisition and representation of preference information which can be seen in Figure 2.10. Some of these methodologies are described in the following subsections.

2.4.4.1 Analytical hierarchy process

Analytical hierarchy process (AHP) was originally developed by Saaty [103] and is a widely used multi-attribute decision support tool. It is designed for the selection of the best from a set of design alternatives by breaking the high-level problem into low-level problems (hence hierarchical) using simple pairwise comparisons. The decision problem creates a pairwise-comparison matrix, $\mathbf{M}$, based on the decision criteria and the relative weighting scale is derived from solving the eigenvector problem

$$\mathbf{M}\ \mathbf{w} = \lambda_{\max}\mathbf{w} \tag{2.19}$$

where $\mathbf{w}$ is the vector of weighting factors and $\lambda_{\max}$ is the highest eigenvalue of $\mathbf{M}$. This allows the set of alternative designs to be ranked such that the one with the highest value is the most favourable. Through the eigenvector approach it is possible to measure the consistency of the pairwise-comparisons, allowing the validity of the answer to be assessed. However, for a large number of decision criteria, it can be difficult for the user to maintain consistency.

AHP can be used to aggregate the opinions in groups via voting strategies or forming the geometric mean as demonstrated in [101]. However, a major criticism of AHP is that it is subject to the phenomenon called rank reversal [104, 105]. This is where the ranking order of the alternatives can change if a new alternative is introduced [106]. Also, as the preferences were captured using a binary scale, the output is also binary. This suggests there is a linear relationship between preferences which is not always the case.

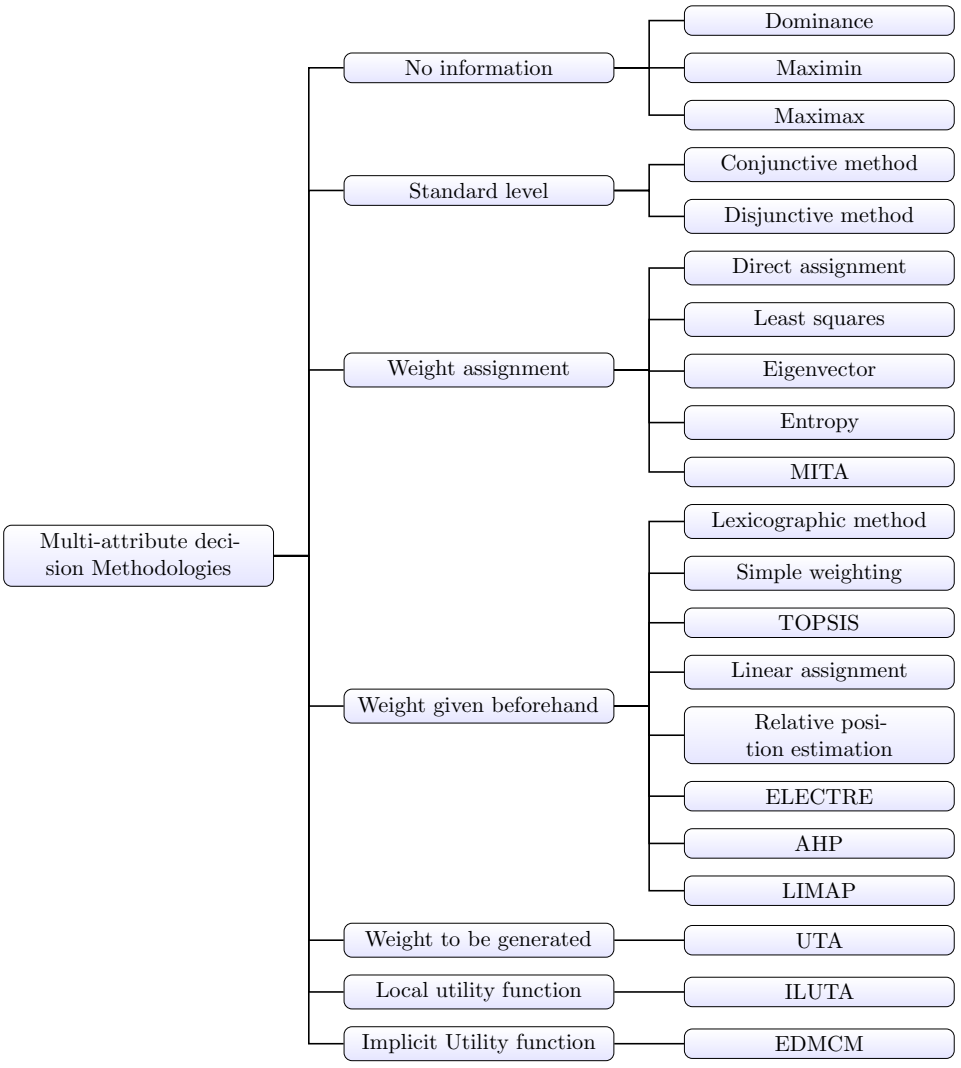


FIGURE 2.10: Classification tree of multi-attribute decision methodologies. Reproduced from Sen & Yang [102]. (Known abbreviations in diagram are as follows: MITA - Minimal Information Trade-off Assessment; TOPSIS - Technique for Order of Preference by Similarity to Ideal Solution; ELECTRE - Elimination and Choice Translation Reality; AHP - Analytic Hierarchy Process; LIMAP - Linear-programming for Multidimensional Analysis of Preference; UTA - Utility Additive.)

2.4.4.2 Multi-attribute utility theory

Multi-attribute utility theory (MAUT) introduces the term utility not yet defined here. In economic theory, utility is defined as a numerical measure of preference relationship. This can be seen as a measure of benefit, satisfaction or usefulness [91] that a system attribute provides. Utility theory (also known as 'the expected utility theorem' by von Neumann and Morgenstern [107]) quantifies decisions made under uncertainty, therefore, takes into account the risk attitude of the decision maker. The utility function, u can be found by presenting the decision maker with a set of probabilistic lottery scenarios (see Collopy [92] for a simple example of the lottery). The scenario is set out as a choice between a certain outcome, say x_i, or playing the lottery with $u(x_B) = p$ and $u(x_W) = (1 - p)$

where x_B is the best outcome, x_W is the worst outcome and p is the probability of getting x_B. From this, a particular probability, p_i, emerges which represents the probability at which the decision maker is indifferent to the choice of x_i or playing the lottery:

$$p_i = \frac{u(x_i) - u(x_W)}{u(x_B) - u(x_W)}. \tag{2.20}$$

The development of the MAUT equation (Equation 2.20) depends on the condition of independence assumed. Keeney & Raiffa [108] demonstrate that a multiplicative form can be used to combine the utility functions as long as the following conditions are assumed or are verified:

1. Preferential independence: implying the preference order between two consequences of an attribute is independent of the level of all the other attributes.
2. Utility independence: implying that the utility function of an attribute is independent of the level of all the other attributes.

If both assumptions hold, then the multi-attribute utility function, $u(x)$, for N attributes is:

$$Ku(x) + 1 = \prod_{i=1}^{N} (Kk_i u_i(x_i) + 1) \tag{2.21}$$

where k_i is a scaling factor in the range $0 < k_i < 1$ for the i^{th} attribute, and K is the multiplicative scaling constant. To find K an iterative process is required as shown in Keeney & Raiffa [108]. Further more, if the additive condition is verified then $\sum_{i=1}^{N} k_i = 1$, the utility function becomes a simple weighted sum of the form

$$u(x) = \sum_{i=1}^{N} k_i u(x_i). \tag{2.22}$$

The advantages of using MAUT are that it can capture uncertainty and risk in the decision making process and allows the decision-maker to establish difficult to make trade-offs between attributes. Because of these advantages and its comprehensive theoretical structure, it has been used in several research and real-world problems (for example, space system design [109] and dairy farming systems [110]).

However, MAUT has its limitations. The first is that the validity of the preferential and utility independence does not always hold in real situations as it asserts a person's preference should vary linearly with the probability of its occurrence which is often found not to be the case. The second limitation is that it is inappropriate when trying to

aggregate the preferences of a group of individuals due to Arrow's impossibility theorem [99]. Finally, Collopy [92] notes that utilities are abstract dimensionless metrics, whose meanings are difficult to comprehend. Therefore, translating them to a certain equivalent worth (*i.e.* a monetary value) is recommended.

2.4.5 Worth based models

Techniques that involve quantifying the monetary value of a system are different ways to rank the alternative designs. Sometimes it seems viable to model the value of a system by defining the value as the system's performance divided by its cost, which avoids assigning a monetary value to the system performance [92]. However, Collopy [92] points out through the use of a simple example this metric generally leads to being incorrect. Therefore, monetising the value of a system is often required.

Net present value (NPV) method is a measure of the profitability of an investment, taking into account the time value of money through the use of discount rates:

$$NPV = D_0 \sum_{t_i}^{t_j} \frac{D(t)}{(1+r)^t} \tag{2.23}$$

where D_0 is the initial investment, $D(t)$ is the future cash flow, t_i to t_j is time period considered and r is the discount rate. The choice of discount rate has a large influence on the final value [91]. However, there is no definite method to select it [42]. This method assumes that the stakeholders only perceive value in monetary terms. The advantages of NPV are that it makes the comparison of design alternatives meaningful and straightforward [11] and the stakeholder's risk attitude can be incorporated [92]. On the other hand, NPV requires all the design attributes to be converted to monetary worth which is a challenging task. Also, NPV assumes all cash-flow and discount rates are known a priori, but in reality they are prone to fluctuations and uncertainties [91].

Cost-benefit analysis (CBA) is another value-centric tool that is useful for quantifying the net benefits yielded by a system to its respective net costs [91]. In the cases where the effectiveness of a UAS mission cannot be readily translated into a monetary value, the goodness of the system can be measured in terms of performance. This, combined with a cost model of the service (acquisition, maintenance and operational costs), is the basis of CBA. The generalised governing equation, where the benefit value (once found) is assigned a monetary value as

$$CBA = \left(\sum B_0 - \sum C_0\right) + \sum_{t=0}^{N} \frac{B(t)}{(1+r)^t} - \sum_{t=0}^{N} \frac{C(t)}{(1+r)^t} \tag{2.24}$$

where B_0 and C_0 are the initial monetary benefit and cost of the system respectively, $B(t)$ and $C(t)$ are the monetary benefits and costs with respect to time in the time period t_i to t_j and r is the discount rate. However, the conversion from benefit to monetary benefit is not required. Instead, the decision maker can be presented with a graph depicting the different levels of mission effective against associated costs [42], allowing the Pareto set of design-alternatives to be identified.

2.4.6 Design decision summary

The use of a VCDM removes the limitations put in place by the requirements set in traditional systems engineering approaches. This increases the design space and also eases changes to the design if the objectives shift. The philosophy behind VCD is to maximise a value assigned to a design through a value function which takes into account the needs of the stakeholders.

Methods like the AHP allow the decision maker to rank the importance of the attributes thus a value can built up using these weightings in the value function. However, this neglects decisions made under uncertainty and the risk attitude of the stakeholder. This can be achieved using the MAUT which allows the individual utility functions for each attribute to be combined. The problem with this, assuming the necessary assumptions hold, is that the output is a dimensionless measure and therefore is hard to understand its meaning compared to the alternatives.

Worth based models, such as the CBA method, overcome the issue with dimensionless measures by representing attributes or performances with a monetary value. However, the act of assigning worth to benefit is in itself difficult and assumes the stakeholder is only interested in monetary value. An alternative to this, where the system's performance is of importance, is the comparison of the system's effectiveness to its cost. Rather than convert the performance metric to a monetary value, it is kept in a meaningful unit and presented graphically to show the effects.

The choice of which model to adopt very much depends on the application of the system and the market environment it falls in, along with the stakeholders' views. This means the choice needs to be made as a design decision with the assumptions and limitations of the model understood [42]. The final considerations for value models are a set of desirable properties presented by Collopy [92]

- Repeatability - once the model is set, if you were to return to it later with the same set of attributes you would get the same value out.
- Transparency - Once a value is given, you need to be able to understand how that value was produced; what attributes made it good or bad.

- Differentiability - If the value function is to be used in an optimisation algorithm, often it will need to be differentiable. Therefore, it is advantageous to build the model from polynomials, ratios and transcendental functions.

Chapter 3

Developing a UAS decision support framework

This chapter presents the development of the framework designed to support decision making and optimisation when designing a UAS service. The decision support framework allows the user to input the specific service definitions (based on the requirements set out in the RFP) and also capture the stakeholders' needs. The weightings on the attributes of the service are then used to find the values of the candidate designs created from the database of COTS UAVs and payloads. To obtain its value each candidate UAS service design is exercised through the operational simulation to gain an understanding of the design's resource usage, mission performance and costs. The accumulated outputs should provide the user with an optimal solution along with other alternative designs. The reasoning for the optimal design should be transparent, such that the user can trace backwards to see exactly how the selected design is defined as the optimal one.

The following sections detail how the framework is built up, what is in scope and the framework's underlying requirements.

3.1 Requirements

For the framework to survive as a tool for those who design UAS services, it needs to be useful to them and meet their requirements. Nurminen *et al.* [111] empirically evaluated several expert systems to determine what made them survive over 10 years of application. The conclusion reached by the study showed that, firstly, as the users are generally experts themselves, the expert-system should complement rather than replace the user. Secondly, the usability of the system should take precedence over automation as usability is considered more important. Finally, the development of the system should be fast and agile to cope with the changing environment often associated with the disciplines

expert systems are found in. However, these findings are based on empirical observations and do not take in to account factors such as the development team's culture and the sensitivity to type of industry or application [111].

Schumann [35] identifies four main requirements, all interconnected, that are demanded from his conceptual design phase, mission-modelling framework: comprehensible and simple, generic, modular and realistic. These are shown in Figure 3.1 with how they relate to each other by either supporting or opposing. These four requirements are also applicable to this framework.

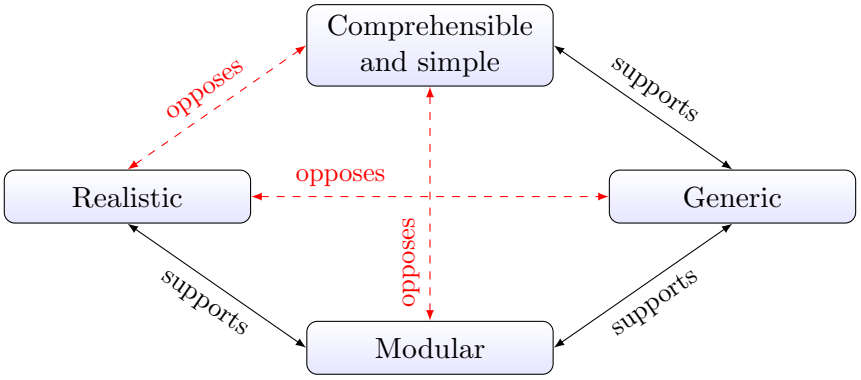


FIGURE 3.1: Framework requirements and interconnecting relationships (either they support or oppose the other requirement). Reproduced from Schumann [35].

In this thesis, the framework needs to be understood by the users. If the user can comprehend how the framework produces its output then they will be better able to judge the validity of the output and how the assigned assumptions and limitations affect it. As the framework aims to support the decision making process, it must integrate into the current methods to provide a smooth transition and be simple to use. It should not require a steep learning curve. However, if it is too simple this can oppose the model's realism and not produce acceptable results.

As indicated in Chapter 1, there are many different applications of UAVs. In addition, there are many different COTS models of UAVs and payloads. The framework needs to be able to recreate these different applications and missions so that the optimal design of the service can be found from the pool of assets. To do this, the framework must be able to accept a variety of different inputs and hence it is required to have generic options that can be applied to build up the specific service model.

To achieve a generic framework that can accommodate many different mission scenarios, the framework design should be modular. In other words, the framework should be compatible with any extensions and add-ons that may be desired later in its life, or required to keep the framework current. This can improve the realism of the simulation. For example, the extension could be an improved payload model or the addition of a communication link model. Also, the ability to modularise the framework based on

physical boundaries is recommended as this can make it more intuitive and easier to adapt.

Finally, realistic results are desired here. However, due to the lack of information and knowledge of the exact CONOPS of the service and the exact performance metrics of the systems, the level of realism should reflect the level of knowledge available. This links to the thinking behind Figure 2.8 in Section 2.3.5, where there is an optimum level of detail with regard to model confidence. Similarly, if the framework produces high fidelity results, but the inputs are not known exactly and are limited, then there will be a lack of confidence in the model as the output should reflect the level of certainty of the input.

3.2 Service framework

With the requirements presented in Section 3.1 in mind, the overall framework is defined in the following sections. To start with, the high level scope of the framework is considered in Section 3.2.1. The scope of the model is also considered at the different levels and is described as the sections progress. The framework is broken down in a similar way as to how the service is modelled. This mostly follows the physical boundaries that form the modular components of the model. These modules comprise of the mission framework (Section 3.2.2), the operating base framework (Section 3.2.3), the UAS framework (Section 3.2.4), the payload framework (Section 3.2.5) and finally the weather framework (Section 3.2.6).

3.2.1 Framework scope

Firstly, this thesis focuses explicitly on the modelling of UAS based services. Although the model produced is capable of introducing different vehicle types (for example a land-based vehicle) to the service by creating a new module, this is not considered here. However, the inclusion of this is discussed in Section 6.2.2 (Application of tool to different UAS service types). Chapter 1 introduced several different types of services currently using UASs. These ranged from military weapon deployment to civilian aerial surveys and maintenance inspections for buildings and structures. The service examples listed were categorised by the user-group, for example military, commercial and governmental applications. However, this categorisation method is only beneficial in terms of visualising the type of task the service may be involved in and not how they are setup. Instead, it is proposed that UAS services are categorised as either *deployed* or *stationed*[1]. A deployed service signifies one where the UAS is transported to a temporary location near (or at) the task location and then activated. On the other hand, a stationed service

[1] *Deployed* and *stationed* are often used to describe the service location of military personnel. In the military context deployment can be considered a temporary location of service, whereas stationed is the more permanent assignment.

signifies the UAS having a permanent location and is activated such that it transits to the location of the task under its own power. This can be thought of as the operating base being *mobile* or *static*. Examples of deployed services would include aerial surveys of individual buildings, entertainment videography and military personal reconnaissance. Whereas examples of stationed services include long range ISR assignments, cargo and transportation, and assisting with SAR or the emergency services. Often, those services that fall under the stationed category conduct frequent missions and there can be multiple missions occurring simultaneously. They also are active as a service for time periods of months to years.

This thesis is concerned with the services that fall under the *stationed* category. These have operating bases that are permanent (for the duration of the service). It is also possible for them to have multiple operating bases to allow the service to cover a particular geographical region.

The UAVs considered within the model are expected to be COTS platforms. This means that the parameters that define the platform (*i.e.* performance values) are known or can be well estimated as opposed to allowing them to be variables. The justification to this limitation is that when responding to a service's RFP the amount of time given by the client to submit and action the tendered service design is often insufficient to take a UAV design from concept stage to production. However, although this is out of scope for the work in this thesis, the framework is capable of supporting decision making based on service performance during the concept stage of UAV design.

Finally, the model aims to allow CONOPS and policies to be defined or set by the designer. For example, a service could have the policy of always flying on full fuel[2] regardless of the expected range or endurance of the mission. However, an alternative policy could be to ensure there is only enough fuel for the expected mission plus a set reserve to allow for potential delays or uncertainty in the expected mission duration. These policies and concepts of operation can have a significant influence on how the service is performed. They can also be unique to the specific service and therefore it is hard to incorporate them all in a generic model.

3.2.2 Mission framework

The mission framework defines when, where and what tasks the UAS platform will have to perform within the service. A service can be built up of any number of mission definitions, each with unique details. The mission definition can be broken down into two main components, the scheduling component and the task component.

[2]Read fuel here as the platform's energy source and therefore this example policy covers both hydrocarbon fuels and batteries, as well as other energy sources.

Firstly, the scheduling component of the mission framework is considered. Services can either have missions that follow a routine where the time of the task is planned, or the task can be random and the mission is reactive to an event or occurrence. An example of the first type is an hourly, daily or even weekly task such as a photogrammetry survey of a coastline looking at the effect of erosion. Whereas, an example of the second is a randomly generated task such as a response to an incident (such as a SAR assignment or monitoring a wildfire). The mission framework allows for both of the scheduling types to be set by the use of probability distribution functions describing the frequency.

The task component describes the *where* and *what* aspects of the mission. Due to the geospatial nature of the missions, the majority of the task component is described geographically. Similar to the scheduling component, the *where* aspect of the task component has two major types. The location is either known and planned, or it is randomly located. In fact, the planned type is just a unique version of the randomly located (*i.e.* where there is no variability in the location). However, in this framework they are classed as different types. An example of a task at a known location is performing an air-drop of medical equipment over a remote clinic. On the other hand, an example of a randomly located task could be, as before, responding to an incident.

So far, the mission location has been described as a location or a point. However, the task (*i.e.* the *what* aspect of the mission) on arrival at the mission location varies in complexity. The type of tasks considered in this thesis are described as follows. Firstly, the most simple task type is a **point** mission. This is where the UAS transits to a location and then returns after completing an instantaneous task such as the air drop example above. Expanding on this type is a **loiter** mission. Here, on arrival to the location, the UAS will loiter for a period of time (*i.e.* remain in the vicinity of the specified location). For a fixed wing UAS this could be performed by flying a circular pattern, for example. The amount of time spent at the location loitering can be fixed or varied. All point and loiter tasks are specified by a latitude and longitude coordinate.

The next set of task types involves a series of waypoints to define the task. The waypoints, given by latitude and longitude coordinates, create a **path** mission. The UAS transits to the start waypoint and then moves to each of the waypoints in the series of waypoints following the sequence. Once the UAS reaches the final waypoint it transits back to its operating base. The series of waypoints can be of any length but the order matters. In addition, the series of waypoints can be predefined, therefore the path coordinates are known in advance, or they can be dynamically allocated (*i.e.* as the mission proceeds). The ability to dynamically allocate the waypoints allows the coordinates to be set based on external parameters. An example application of this is the creation of search patterns or surveys (*e.g.* an expanding square search pattern) where the swath width on the ground is required to overlap slightly to ensure full coverage of the area. The **dynamic path** is the final task type considered in this thesis. However, it is possible to adapt the model and add new types as they are required.

By setting out the types of tasks that the UAS can perform during a service the importance of the geospatial modelling becomes apparent. The importance is demonstrated when the locations are randomly generated as often they need to follow some form of distribution but also remain within the region the service is covering. An example of this could be a service for monitoring and responding to wildfires. In this scenario, the wildfires are likely to follow a random distribution with areas of higher likelihood over particular vegetation types. However, they are not expected to be found over the sea. Also, the stakeholders may only have permission to operate in a particular area restricted by, for example, political boundaries. This thesis introduces these geographical region as either **include polygons** or **exclude polygons**. One way to visualise this is to treat the *exclude* polygon as a hole in the *include* polygon. There can be multiple *exclude* polygons in one *include* polygon. A polygon is defined by a list of latitude and longitude coordinates, similar to the path task type described above. However, the polygon differs as the list must form a closed loop, encapsulating the region.

3.2.3 Operating base framework

An operating base is defined as a static, geographically located site that has UAS and resources assigned to it. As described in Section 3.2.1, this thesis focuses on static operating bases with locations that are permanent for the duration of the service. This allows the resources associated with the UAS to be situated there, for example, the maintenance equipment, consumables and refueling equipment. It is assumed that the operating bases are capable of facilitating the launch of the UAS types that are assigned to it. Each operating base is assigned a daily cost to account for the rent and use of the facilities.

The operating base can be assigned any number of UASs and any number of UAS types. Therefore, it is assumed that the operating base has been selected such that it can store the amount assigned. It is also assumed that each UAS assigned to an operating base is operated exclusively by the personnel at that operating base. This means that the UAS will require the personnel at its operating base to be available before it can fly. It can not use personnel from a different operating base, nor can it land at a different operating base. This does introduce a few restrictions to some of the CONOPS that might be considered. For example, if an in-flight emergency occurred, the UAS will return to base rather than seek out an alternative (potentially nearer) operating base. However, by restricting this type of operation, it means each operating base will always have access to the UAS assigned to it. Also, it means that the operating base is always suitable for the UAS to take-off and land (via the assumption laid out in the above paragraph). Finally, it also removes the need to include several complex logistical methods to deal with these scenarios, for example, in the case of the emergency landing, the logistics of returning the equipment to the original operating base.

The operating base also has personnel assigned to it. Here, this is demonstrated by the assignment of pilots. However, the inclusion of other personnel, such as engineers or technicians can also be accommodated. This would just require integrating their interaction with the service and equipment into the model. Alternatively, if these roles interact with the service in the same way then they can be combined and 'a personnel' can be thought of as 'a crew' required to operate the platform. In the case of pilots, each operating base can set the maximum number of pilots assigned to it. Then the number of pilots at the operating base follows the number of UAS units assigned to it up to this limit. This allows the effect of personnel limitation to be studied.

The basic setting for personnel assigned to an operating base is that they are assumed to be available 24 hours a day, 7 days a week. A more complex model can be implemented that considers the working hours and shift patterns of the personnel. Although, while experimenting with applying this complex model it was found to introduce unnecessary complexity and policies with regards to shift handovers. Therefore, the basic setting is deemed acceptable and can account for these ideas through different methods. For example, if the missions are only scheduled to occur during the working hours of the personnel, this would be accurately represented. Alternatively, if the missions are randomly scheduled to occur at any time of the day, the salary of that personnel role can be set to account for the number of people required to cover the availability.

3.2.4 UAS framework

The UAS framework captures the performance and the reliability of the platform. It also contains the platform's state throughout the service simulation. The UAS model presented in this thesis is focused on fixed wing platforms as justified in Section 2.2.2. Currently, the performance models of the UAS covers both internal combustion engine (ICE) and battery powered platforms using the equations described in Section 2.2.4.1. These equations require knowledge of the UAS's aerodynamics, mass and propulsion system. It is assumed that these values are available or can be well estimated. These two performance equations allow the UAS model to account for weight change due to fuel being burnt in the case of the ICE powered platforms, and account for the effective capacity of batteries in the case of electric powered platforms.

By using the performance equations described in Section 2.2.4.1 (Equations (2.3) and (2.9)), the UAS range and endurance can be found. These values are used to determine if the platform can attend the mission and how much it can complete before having to refuel. However, it is not just energy supply that limits the range of a UAS in the real world. There are two other factors that come into play; the communication range and the legal range. Firstly, the communication range is the distance after which the command and control link to the platform is no longer effective. This is often calculated using the link-budget equation which takes into account transmitter power, antenna

gains, hardware losses, miscellaneous losses and importantly the path loss. The path loss component contains the terms for the wavelength of the radio frequency used and the distance between the transmitter and the receiver. The link-budget equation can be arranged to find the distance at which the system can be deemed reliable (*i.e.* setting the received power to an appropriate margin to account for real-world noise, for example 10 dB). Although this is an important consideration, it was decided to be out of scope for the model presented in this thesis. Instead, it will be assumed that the UAS will be equipped with a communication system that is capable of a reliable link at the maximum range the UAS can travel before returning to base. To compensate for this limitation, the cost associated with the UAS flight hours can be adjusted to account for the communication system used.

The second factor was the legal range. This is the legally permitted distance between the UAS and remote pilot in command set out by the governing aviation authority (*e.g.* the CAA in the UK). Current civil regulations in the UK, for which guidance is provided in CAP 722 [27], restrict UAS general operations to 500 m from the remote pilot in command without any additional exemptions. However, the Basic Regulation [112] (on which the civil regulations are based) lists aerial activities to which these regulations do not apply. Within this list are aircraft carrying out military, customs, police, search and rescue, firefighting, coastguard or similar activities or services (which are known as 'State aircraft') with the caveat that the state must 'ensure such services have due regard as far as possible to the objectives of the Regulation' [27]. As the listed services align with the targeted services of this model the legal range is not included as a UAS limitation. In fact, this model could be used to gain an insight into the expected range of the UAS for the optimal service design and use this data to build and strengthen an operating safety case. However, the financial implications associated with the creation and maintenance of the operating safety case and legal-based cost need to be considered. This was taken into account through the cost per flight hour.

The reliability aspect of the platform model is broken into two main elements. The first is a system failure which leads to maintenance, and the second is a system failure which leads to a critical failure and total loss of the platform. Both failures use flight hours as a measure of when a failure will occur. The time-to-fail values are generated on creation (and replacement) of the UAS platform and are randomly selected from a Weibull distribution. Weibull distributions are commonly used for modelling reliability performance of systems [113] [114] and the basic distribution can be described using two parameters, α and β where α is the shape parameter and β is the scale parameter. β can be set if the mean, μ, and shape parameter are known by rearranging

$$\mu = \alpha\Gamma\left(1 + \frac{1}{\beta}\right) \tag{3.1}$$

to find β (note that the gamma function[3], $\Gamma(x)$, will produce two possible values of β and therefore care needs to be taken to ensure it is compatible with the mean). This model treats the system as a whole in terms of time-to-failure and therefore the Weibull parameters should reflect this.

The two reliability elements differ with regards to how the platform reacts to a failure. In the case of a critical failure the platform is considered a total loss and a replacement is required. On replacement a new time-to-fail for the critical failure is generated and the process is repeated. The length of time to complete the UAS replacement can be set (*e.g.* this could be instantaneous or require days to get the replacement operational). No other factors influence the time-to-fail and therefore the parameters of the Weibull distribution should reflect the reliability of the platform, the in-built redundancy of components, and the effect of the maintenance programme[4] on the active lifetime of the platform.

On the other hand, when a system failure occurs, the platform is able to return to base but requires immediate maintenance. Once the maintenance operation is complete, a new time-to-fail value is generated and the UAS is made available for operations again. The model also includes a basic maintenance schedule option that can be turned on and represent preventive maintenance, or turned off to represent run-to-failure maintenance. If in preventive maintenance mode, an extra parameter is required to state the number of flight hours between scheduled maintenance. This is often set such that it is just before the mean-time-to-failure.

The reliability and maintenance policies are simplifications of the real process. However, for the fidelity of the model, this level of simplification is deemed acceptable. It also allows for a simple study into the effect of varying the maintenance policies should this be of interest.

3.2.5 Payload framework

The payload framework presented as part of this model and thesis focuses mostly on electro-optical (EO) sensors as described in Section 2.2.5. The justification for this is that most services that are covered by the scope of this model use electro-optical sensors. The payload can be modelled as an inactive payload (*i.e.* a constant weight for the duration of the mission), a releasable payload (*i.e.* medical aid delivered by air drop) or a stabilised static electro-optical sensor (*i.e.* a forward and down facing camera stabilised in pitch and roll). These assumptions dramatically reduce the complexity of the model and allow it to still capture the importance of the payload functionality. This also allows the study of payload choice.

[3]The gamma function expands the domain of factorials to non-integer values

[4]This refers to those maintenance programmes suggested in Section 2.2.3, such as daily inspections or replacement of life-limited components.

For services that require target detection, work was put in to implement a model that can set the flight regime to achieve a particular NIIRS rating of the expected targets (*e.g.* search and rescue victims). However, eventually this was deemed out of scope due to the complexity of predicting NIIRS performance and the lack of data on the parameters required for the general image quality equation (Equation (2.18) presented in Section 2.2.5). Instead, the Johnson criteria described in Section 2.2.5 was used which relies on more tangible parameters for the EO sensor.

3.2.6 Weather framework

The weather framework introduces a real-world variable to the model. In this thesis, the weather framework contains information about the temporal wind. However, through the modular design further weather categories can be added, for example visibility which would have an impact on the payload's functionality. Wind was selected as it has a significant effect on the range and performance of a UAS. By applying a global wind layer to the geographic region and varying it with time to match the historic distributions, a more realistic model is produced.

The wind component models the wind using historic values for the wind direction and wind speed to build custom distributions for each month of the year. These distributions can then be sampled and the resultant vector can be used to calculate the effective range of the UAS. Also the ground speed and flight distance can be adjusted accordingly.

The assumptions made by using this method are that the wind vector is constant between sampling times and is uniform across the whole region modelled in the service. Also it is assumed there is no hysteresis in the vector (*i.e.* the next sample is not affected by the current sample). This simplification can lead to large step changes in the wind conditions which can be detrimental to realism of the model.

The wind direction data set should consist of a set of compass headings and the probability the wind came from that heading for each month of the year. Then, through linear interpolation, a custom distribution can be built for each month. The more compass headings available in the data set the better.

The wind speed data set should consist of a set of monthly wind speed distributions. Alternatively, if the distribution data set is not available it is possible to use the monthly average wind speeds recorded at 10 m above the ground. This can then be applied to a Rayleigh distribution (a special case of the Weibull distribution, where the shape parameter $\alpha = 2$) such that the scale parameter β can be set using the mean μ by the use of the following equation

$$\beta = \frac{\mu}{\sqrt{\frac{\pi}{2}}}. \tag{3.2}$$

This allows the variability in wind speed to be modelled when the shape of the distribution is not known. Several studies into wind speed distribution predictions have been presented, mostly due to the rise in using wind as a renewable energy source and the need to assess wind conditions and the suitability for placement of a wind turbine [115] [116] [117]. In these studies multiple different distributions are considered and compared but often the shape parameter of the distribution is calculated from the existing data set. Therefore, if only the mean wind speed is known, the use of a Rayleigh distribution is sufficient.

Wind speed also varies with altitude so a wind profile was applied. This allows the altitude of the UAS to scale the wind speed appropriately. The log wind profile [118] was selected and is applied with the following formula

$$u(z_2) = u(z_1)\frac{\ln(z_2/z_0)}{\ln(z_1/z_0)}, \tag{3.3}$$

where $u(x)$ is the wind speed at altitude x; z_1 is the reference wind speed altitude (here 10 m); z_2 is the UAS's altitude and z_0 is the roughness length to account for the effect of the roughness of a surface on wind flow. z_0 should be set to match the terrain type (*e.g.* grassland typically has the range 0.01 to 0.05). The log profile was selected over the power law profile due to the ability to fine tune it to suit the terrain type.

3.3 Framework summary

This chapter presented the underlying requirements to improve the chances of this framework being adopted as a useful tool in supporting UAS service design. Also presented were the assumptions, simplifications and limitations made in the formulation of the model and used in this thesis. The reasons and justifications were provided along with the effect they have on the output of the model.

Some limitations were introduced to remove complexity from the model, thus allowing the entire framework to reach a status acceptable for testing and to analyse its effectiveness. However, these limitations can be lifted, due to the modular nature of the framework, by the creation of additional modules or more complex logic flow.

Other assumptions and simplifications were made due to the availability of data for what is being modelled. Where data values for a model component were lacking or hard to obtain from published data, simplifications were made to ensure the overall model was still able to capture the effect of the component, albeit to a lesser degree of accuracy. However, one benefit of the simplifications is that a lower level of detail can improve the model confidence as discussed in Section 2.3.5.

Chapter 4

Simulation and modelling

This chapter presents the simulation and modelling methods for finding an optimal UAS service design. Firstly, the simulation design is discussed in Section 4.1. Section 4.2 gives an appraisal of the tools considered in the formulation of the model. Finally, Section 4.3 demonstrates the implementation of the model.

4.1 Design

Based on the literature review and the problem being modelled, an event-driven simulation was chosen as the basis of this model. The simulation uses a hybrid of discrete event modelling and reactive agent based architecture. The model is represented in the form of a high level flow diagram shown in Figure 4.1. There are three main areas to this diagram: the user inputs, the simulation and the framework outputs. The components outside these areas are either processes working in the inputs, or storage of data.

The inputs are taken from the RFP as requirements and service details. The requirements contain information about what the customer and stakeholders desire and these can be ranked using methods described in Section 2.4.4.2. This leads to weighted preferences, for example the optimal design candidate should prioritise completing the missions over the cost. Finally, these weighted preferences contributes to the formation of the value function using a cost-benefit analysis method described in Section 2.4.5.

The service details outline what the missions comprise of, such as geographical points of interest (*e.g.* the operating base locations, geospatial polygons). The service details also outline the probability distribution of mission attributes that define the service (*e.g.* frequency, type, location, duration). These data sets are then used to model the missions in the service planner. The time period of the service is also defined within the service details so that the simulation has a start and an end event. Also included are any global CONOPS or policies that are expected to be followed in the service model.

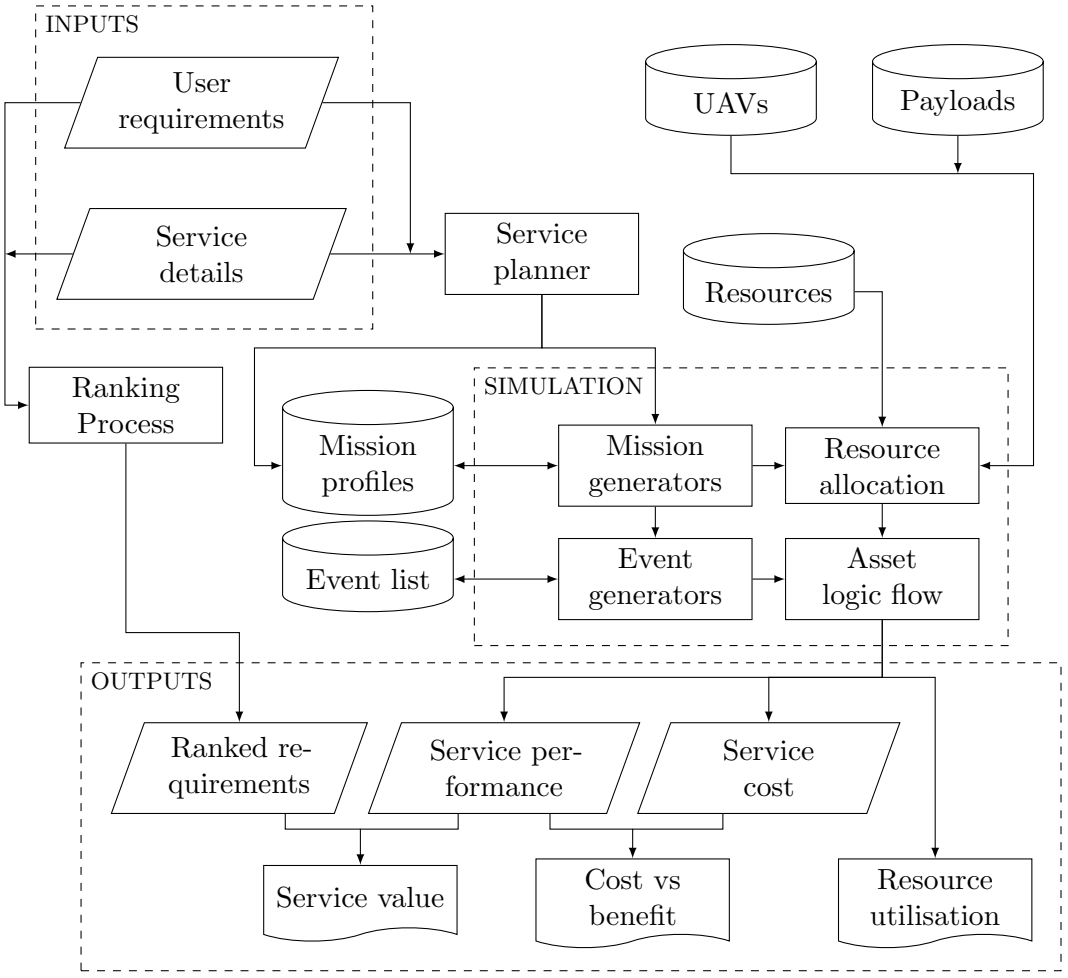


FIGURE 4.1: A high level flow diagram of the framework highlighting the user inputs, simulation area and outputs. The components outside the dashed boxes are either processes working on the inputs (for example, preparing it for the simulation), or storage of data (for example, the database of UAV details). The arrow directions show the flow of data or information, ultimately arriving at the service value, CBA and resource utilisation.

The UAVs and payloads have their details stored in external databases. These details are accessed by the simulation to model them in the service using methods described in Section 2.2.4. The information stored in these databases include performance metrics, reliability data, unit costs and operational costs. The simulation also calls on a collection of databases that contain information on the required resources, for example, personnel details such as pilot salary, and consumable information such as fuel price. The simulation can be set such that the service has access to infinite resources. This allows the designer to evaluate how well the service can perform without any resource restrictions. Alternatively, it can be set to restrict the resources to see the effect resource limitations can have on the service, for example the number of pilots available.

The service planner translates the information provided in the service details and the user requirements to create the mission profiles. These, in turn, define the inputs for the generators that produce events during the simulation. The key generators within the

simulation are the "mission generators". These generate the events that request a UAS (and the required resources) to become active and fulfil the mission's objectives. Once a UAS has been requested it follows through the asset logic flow based on the generated event (such as take-off, transit, waypoint navigation, landing, refueling, maintenance, etc.). Within the asset logic flow, and while the UAS is active, other events (via the event generators) can still be generated that affect the state of the UAS (for example, a change in weather conditions).

Finally, once the simulation has run for the stated time frame, it outputs the performance of the candidate design under consideration and the cost generated from the simulation along with the resource utilisation. These, combined with the ranked requirements, allow the service value and cost benefit analysis to be calculated. These values are used to compare the service candidate designs and find an optimal solution.

4.2 Tools

During the early stages of the research, several software tool kits (STKs) were investigated to see if an off-the-shelf program could usefully provide a computational environment for the model and simulation. These tools ranged from aircraft simulators (which provided realistic flight dynamics and environments) to computational simulation suites (for the design of computational models). There were two areas tools were considered for:

1. A mission simulation environment in which a 3D representation of the mission can be shown and data collected regarding the sensors and UAS performance (see Section 4.2.1).
2. A programming environment in which the service model can be designed and preferably from which the mission simulation environment can be accessed (see Section 4.2.2).

Several key aspects of the software were considered, such as: is a licence required, is the software open-source or code-programmable, is the interface a graphical user interface (GUI) or a command line interface (CLI), can the simulations be automated, and which aspect of the project does the tool aid. A review of the state-of-art software for agent based modelling by Abar, Theodoropoulos *et al.* [119] was also useful for shortlisting those being considered under the agent-based category. The following sections present a brief overview and evaluation of each of the STKs that were investigated.

4.2.1 Mission simulation tools

4.2.1.1 AGI STK

AGI STK (Analytical Graphics, Inc. Systems Tool Kit) is a four-dimensional (3D space and time), physics-based simulation environment for testing land-, sea-, air- and space-based systems [120]. Example use-cases are simulating satellite positioning, radio frequency (RF) communications and radar modelling, aircraft flight modelling and missile modelling. Aircraft missions can be created and flown in the 3D graphical world to evaluate the performance of payload sensors and communication equipment while also monitoring the aircraft's performance. Many of these features are appealing for the design of a UAS service because they allow visualisation of the mission performance relative to the RFP.

However, although the base tool kit is free to download and use, any add-on modules require the purchase of a licence. To benefit from the power of AGI STK several add-on modules would be required, for example: *Aviator* or *Aviator Pro* - to provide enhanced aircraft performance-characteristics and route-modelling; *Integration* - to automate repetitive tasks from outside the STK application though the use of scripted languages to manipulate the application programming interface (API); and *Analyzer* - to allow parametric studies and probabilistic analysis [121].

Moreover, the suite produces high fidelity results from its time-stepped game-like graphical mission simulations. This could potentially require a large amount of computational resources especially when run multiple times. Nonetheless, it is a very capable tool for mission creation, analysis and operation. It could be a useful tool to use once an optimal candidate design is generated via a lower fidelity model to verify the findings or continue designing lower level decisions.

4.2.1.2 Presagis STAGE

Presagis STAGE is a high-realism simulation environment. Its purpose is to create sophisticated simulation scenarios aimed at military operation testing and training, and virtual mission rehearsals [122]. The system uses gaming graphics, high fidelity vehicle models and complex terrain databases to create a dynamic virtual environment with high levels of detail.

The immersive gaming graphics are impressive and would support training of personnel in battlefield decision making and mission tactics. However, this quality of graphics and realism is not essential for the mission simulation aspect and could be considered inefficient use of computational resources. Also, STAGE requires an expensive licence to run.

4.2.1.3 Ternion FLAMES

Ternion FLAMES (Flexible Analysis Modelling and Exercise System) is a similar simulation software tool to Presagis STAGE. It offers a simulation environment for testing, evaluating and analysis of systems; training personnel and operators; and mission planning and rehearsals [123], with the target audience being the aerospace and defence sectors.

FLAMES was used by Cassidy *et al.* [124] in a conceptual design study of a single military aircraft by integrating traditional design analysis and optimisation with battlefield simulation techniques to assess the impact of performance parameters on mission effectiveness [124]. Cassidy *et al.* comment that the determination of fidelity levels was an important factor. For simple manoeuvres the built-in movement models were sufficient at modelling turn-rates and angle of attack (AOA) conditions. However, the more complex manoeuvres (air-to-air combat and surface-to-air missile avoidance) required the incorporation of a more accurate model. Again, this high fidelity and quality of visualisation is not essential for the model being designed in this thesis.

4.2.1.4 NASA World Wind

NASA (National Aeronautics and Space Administration) World Wind is an open-source 3D world model that takes satellite images, elevation data and other geographical information for users to visualise, manipulate and analyse data in a virtual globe representation [125]. This software development kit (SDK) has been used to monitor weather patterns [126], visualise earthquakes and their depths [127] and track satellites orbiting around Earth [128].

The World Wind application is operating-system independent and can be created as a desktop application, a web application or even as a mobile-device application. The geographical rendering is taken care of through the application, leaving the user to build their own geospatial components and models. Due to the open-source nature of the application, it offers a lot of freedom and is designed to be extensible. However, it requires knowledge of coding languages (Java for the desktop application, hyper text markup language (HTML) and JavaScript for the web based application) as well as an understanding of their API.

4.2.2 Programming environments

4.2.2.1 AnyLogic

AnyLogic is a tool kit that supports agent-based simulation (ABS), discrete-event simulation (DES) and system dynamics. The software uses a Java environment to

allow operating-system independent development of the model. The use of a Java environment also means that a runtime licence is not required for simulations as applets and applications can be created. Although there is a free *Personal Learning Edition* licence, this comes with a few limitations (most notably on the model size). However, this can be resolved with either a *University Researcher* licence or a company *Professional* licence.

The tool kit has a clean interface and allows the model to be graphically animated for better visualisation of the simulation. The target audience differs dramatically from the mission simulation tools such as AGI STK. AnyLogic is aimed at users who want to focus on the logic of the problem as opposed to creating a graphically-realistic simulation environment. Because of this, AnyLogic has a wide range of applications from modelling passengers' movement around airports [129] to the logistics of managing rail yard capacity [130].

AnyLogic was used in Schumann's thesis [35] which used UAS life-cycle mission modelling to aid the conceptual design process. AnyLogic provided the base for Schumann's framework and required some work to get a geographic information system (GIS) to model the UAS's mission spatially. Now, AnyLogic has a built-in GIS method to facilitate, for example, supply chain transport and delivery route modelling.

4.2.2.2 GAMA Platform

GAMA Platform (GIS Agent-based Modelling Architecture) is an environment for developing spatially explicit agent-based simulations [131]. It uses its own high-level agent-based language, GAML (based on Java), to allow the user to create models across multiple application-domains. Examples are given for transport, urban growth, epidemiology and environment domains. GAMA can be used for large-scale simulations and the user interface facilitates different 2D and 3D simulation views and allows the user to monitor individual agents within the simulation.

The partnering of an ABS environment and the ability to input GIS data in a stand alone platform is appealing for this project. However, its limitation is that it is only able to create agent-based models and not other simulation methods such as discrete-event models or a hybrid model.

4.2.2.3 ExtendSim

ExtendSim is a powerful simulation tool in which the user can develop dynamic models to study relationships and find an optimum solution [132]. ExtendSim can be used to create continuous, discrete-event and discrete-rate models as well as Monte Carlo, agent-based

and state modelling approaches. ExtendSim was used by Yu [7] in the development of a hybrid agent and discrete-event model of aircraft engine fleet maintenance.

This software requires a licence to build and save models, but once the model is complete an 'analysis run-time' licence can be purchased to run the model for experimentation and optimisation. An evaluation of DES software, using an evaluation and selection methodology by Tewoldeberhan *et al.* [133] in 2002, put ExtendSim in a competitive shortlist because it scored well against the criteria of a company's simulation team.

4.2.3 Tools summary

An overview of these tools can be seen in Table 4.1. The majority of the mission simulation tools investigated came with expensive licences. Also, when noting the level of detail presented in the RFP (that form the service details), the simulation tools had unnecessary levels of detail and fidelity which, in turn, would add a computationally expensive footprint. The exception to this was the NASA World Wind application which was both open-source and highly adaptable via code.

The programming environments highlighted a few useful tool kits. However, as this framework aims to use a DES for part of the model, the GAMA platform has to be ruled out as it only offers agent-based models. This left both the AnyLogic and ExtendSim COTS software tool kits to be considered as potential candidates.

Another option for the programming environment, which was not explored above, was to use a library within a coding language. For example, SimPy[1] [134] is a process-based DES framework based on standard Python [135] and allows simulations to be created within a script or Jupyter Notebook[2]. This option would allow full control over the design of the model and possibly ease the integration with other applications should this be required. However, coding all the components from the ground up and integrating them would increase the work load significantly. It also may fall short of the capabilities of a COTS software package in terms of verification via animation and quick prototyping in a tailored software environment with existing code-blocks (*i.e.* predefined objects for DES models).

4.2.4 Tool Selection

Based on the results of the tool research, the majority of the tools were tried out. Attempts were made to utilise the open-source tools (for example, NASA WorldWind was integrated with a Python-SimPy script to feed inputs to the Java-based world

[1] For more information about SimPy go to `https://simpy.readthedocs.io/en/latest/`.

[2] Jupyter Notebook is an open-source web application that supports over 40 languages and allows the user to create interactive code, equations, visualisations and explanatory text (see `jupyter.org`).

Tool	Free	Code models*	GUI	Comments
AGI STK	✗	✓**	✓	Expensive due to multiple licences for individual modules. Requires an add-on module to automate using a script.
Presagis STAGE	✗	✓**	✓	Expensive and unnecessary levels of 3D-graphical realism. Tool aimed for a different market.
Ternion FLAMES	✗	✗	✓	Tool has high fidelity 3D graphics. Requires licence for custom simulations and additional applications.
NASA WorldWind	✓	✓	✗	Open-source Java and web-based program with 3D world model. Suitable for mission modelling.
AnyLogic	✗	✓	✓	To make full use, a licence is required. Supports both agent based modelling and discrete event simulations.
GAMA	✓	✓	✓	Designed for building spatially explicit agent-based simulations. It uses its own high-level language, GAML.
ExtendSim	✗	✓	✓	A multi-method simulation suite with a clean graphical user interface.

TABLE 4.1: Comparison of software tool kits considered for this project. *The *Code models* column covers both tools that are open-source code and tools that allow you to program the model in a programming language. **The AGI STK Integration add-on and Presagis STAGE Pro version allows programming and automation.

model). However, laying the foundation and communication channels between the open-source applications was time consuming and prone to introducing errors which would affect the entire framework. Also, as discussed in the comments on simulation methods (Section 2.3.5) and highlighted in the framework requirements (Section 3.1), the model would greatly benefit from further work to make a user-friendly interface as most the inputs were through code or command-line operations. Again, time was spent pursuing the open-source route for the graphical user-interface framework which encapsulated all the other modules, databases and components. However, it became apparent that using an existing COTS software tool kit would be very beneficial in terms of rapidly prototyping ideas and handling the animation and user interface components.

Therefore, AnyLogic was selected as the software to provide the programming environment. This was due to it supporting agent-based models, discrete event simulations and GIS integration. It also allows for a high-level of customisation due to being Java-based. However, all the work previously done in the open-source Python code provided a deeper understanding of how to formulate a DES from the basic components and was very beneficial early work. The AnyLogic model also provides a graphical view of the simulation environment which aids the model building process as the user can perform a visual check that the model is behaving correctly when running but also turn this off for the batch runs.

4.3 Simulation components

The following sections demonstrate the implementation of the model through the AnyLogic software. The layout of this section aims to present the implementation of the model in a logical order. To do this, Section 4.3.1 starts with the high level discrete event model of the mission processing. Then, this is followed by the state-chart logic of the UAS agents in Section 4.3.2. The objects, events and agents that interact with the UAS are also presented.

4.3.1 Main discrete event model

The main discrete event model is based around the generation and processing of missions. This is because all the service types this framework aims to cover are based around the UAS attending a mission. Figure 4.2 captures the main events in the mission processing. These are in the left hand column titled *event flow*. Also included in Figure 4.2 are the high level actions of the assets that are used to process the mission. These are found in the middle column titled *asset action*. Finally, Figure 4.2 shows in the right hand column the inputs that are required to describe the assets, the actions or the events.

Working down the left hand column, the first item is *create simulation environment*. This is where the assets are placed in the simulation environment and allocated to their respective operating bases using the service details and the particular design candidate's variables. Firstly all the operating base agents and the mission generator agents are created from the data provided in the service details. This includes the assignment of personnel to each operating base. Other event generators, such as the weather generator, are also created at this stage. Next, the UAS agents are created and assigned. This is achieved through a function at startup which creates a blank UAS agent for each of the platforms in the design. These blank UAS agents are then assigned the details of the platform they are modelling and the details of which operating base they are assigned to. This method places the pool of UAS assets on the same hierarchical level as the

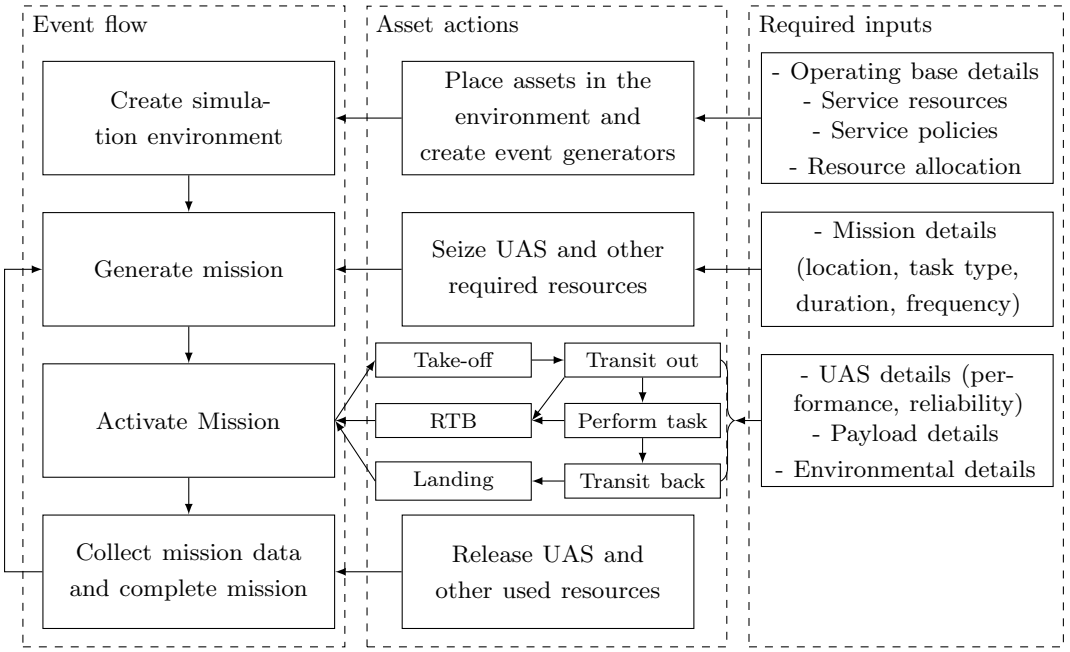


FIGURE 4.2: A simplified flow diagram of the simulation top level events relating to the mission object. The corresponding asset action for each event flow item is displayed along with some of the required inputs for each stage. This can be considered as expanding on the *simulation* section of the diagram presented in Figure 4.1.

operating bases and generators. The advantage of this is that it allows mission-resource allocation to easily search the entire pool of UAS agents rather than having to collate each operating base's pool of UAS agents and then search through that. It also unlocks the ability to reassign with ease a UAS to a different operating base if required in future development. Figure 4.3 shows a simplified diagram of the model hierarchy to clarify this.

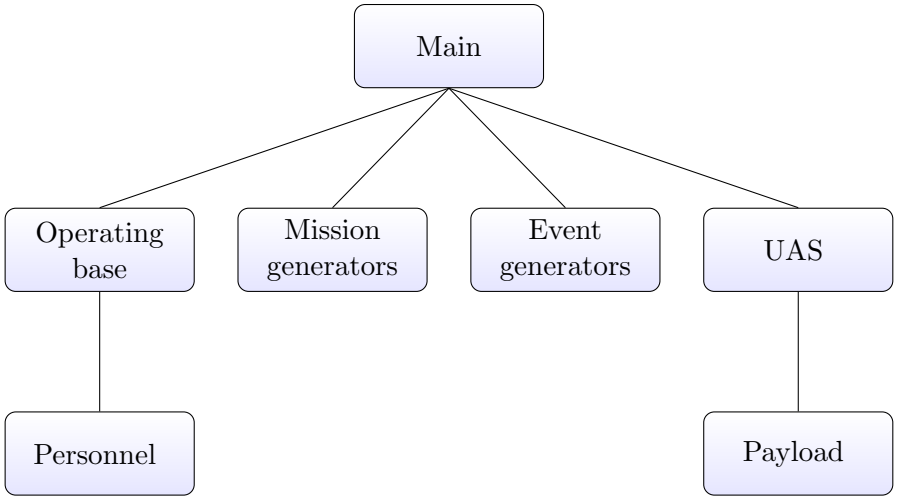


FIGURE 4.3: Diagram of the model hierarchy.

The mission generator agents contain all the information regarding the scheduling, location and task type of the missions. The location of the mission can be generated from

either a known latitude and longitude coordinates (*i.e.* following a data set) or sampled from a probability distribution. There are two types of probability distributions created in this framework. First is the regional distribution where a random location within a geographical polygon (an include polygon) is selected. Exclude polygons, as mentioned in Section 3.2.2, can be applied to the polygon which can be thought of as holes in the sample area. An example of this is shown in Figure 4.4. When the simulation is run in animation mode, the polygons are made visible to confirm correct model setup. The include polygons are displayed as transparent green polygons and the exclude polygon as transparent red.

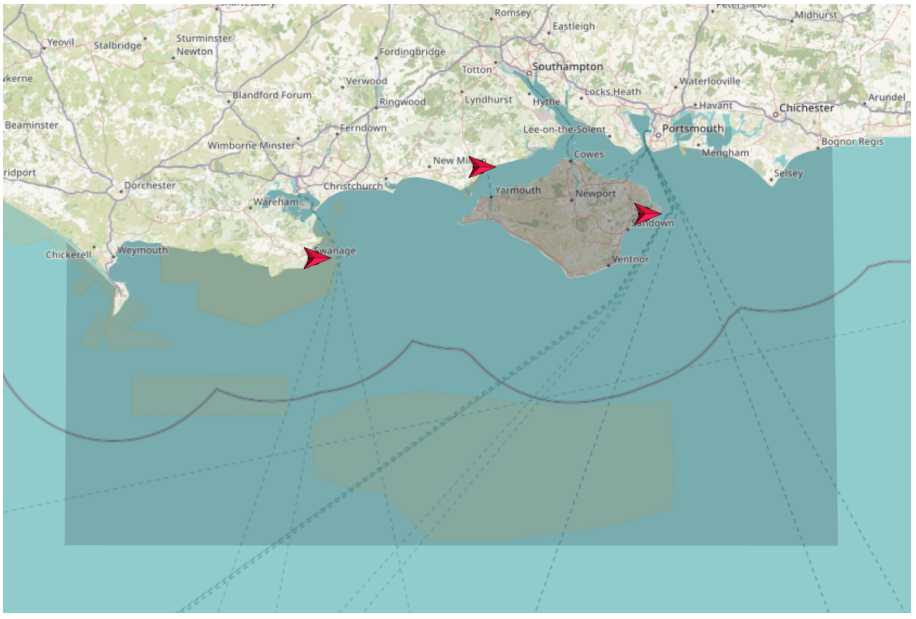


FIGURE 4.4: Regional distribution for mission generation as seen in the AnyLogic simulation. The transparent green polygon following the coastline and out into the English Channel is the include polygon. The exclude polygon follows the coastline of the Isle of Wight and is displayed as a transparent red polygon. The red symbols are UAS agents positioned at operating bases.

The second distribution method is an arc-based distribution. This is where an arc is formed around a longitude and latitude coordinate. This arc has a radial minimum and maximum as well as a bearing start and finish[3]. Mission locations can then be generated using two distribution functions to determine the bearing and radial distance from the centre. For example, the locations could have a uniform bearing distribution and a triangular radial distribution (potentially with truncation if the distribution's limits exceed the arc's limits). An example of an arc-based distribution is shown in Figure 4.5. This example demonstrates the assignment of both radial minimum and maximum and bearing start and stop values. Exclude polygons are also applicable and are in use in this image (note that they are hard to see in Figure 4.5 as they cover all the land regions so there is no obvious distinction).

[3]The order of the start and finish is important as the arc is swept in a clockwise direction.

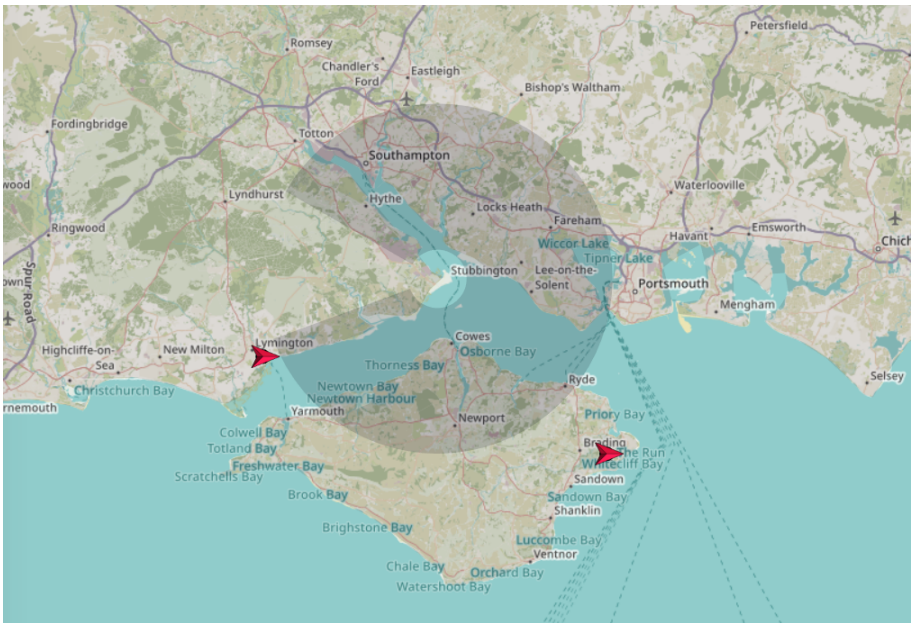


FIGURE 4.5: Arc-based distribution for mission generation as seen in the AnyLogic simulation. The transparent green arc is the region in which missions will be generated based on a radial and bearing distribution. Exclude polygons are also used, in this case two are used, one over the mainland and one over the Isle of Wight. The red symbols are UAS agents positioned at operating bases.

As the model is dealing with latitude and longitude coordinates, all locations generated by the arc-based mission location distribution use the great-circle distance calculations. This assumes a spherical earth and ignores ellipsoidal effects. For example, given the arc's centre location, the sampled bearing and the sample radial distance the location of the mission can be found by

$$\phi_2 = \arcsin\left(\sin\phi_1 \cos\delta + \cos\phi_1 \sin\delta \cos\theta\right) \tag{4.1}$$

$$\lambda_2 = \lambda_1 + \arctan2\left(\sin\theta \sin\delta \cos\phi_1,\ \ \cos\delta - \sin\phi_1 \sin\phi_2\right) \tag{4.2}$$

where ϕ is latitude, λ is longitude and the subscripts relate to arc centre (1) and mission location (2), θ is the bearing (clockwise from north), δ is the angular distance ($\delta = d/R$ where d is the radial mission distance from the arc's centre and R is the earth's radius).

The mission generator agents also contain an event trigger. The event trigger is scheduled based on the settings provided in the mission details input. Once triggered a mission agent is produced and passed to the DES process flow in the main program of the simulation. The creation of a mission agent is the *generate mission* block in Figure 4.2.

The DES process flow for the mission agent is shown in Figure 4.6. This image is a screenshot of the graphical editor in AnyLogic. It can be read from left to right starting

at the *mission_enter* block, where the mission agent enters the process flow. The next block the mission agent reaches is the *seize_uas* block. This is where the UAS is allocated to the mission agent. If there are no UASs available then the mission agent will queue until either a UAS is made available or the mission agent reaches its shelf-life time (*i.e.* if the mission is time sensitive and has failed to be allocated a UAS within a certain time frame). If the latter occurs, the mission agent leaves the queue via the top graphical port and is recorded as a timed-out mission at the *timeout_mission_exit* block.

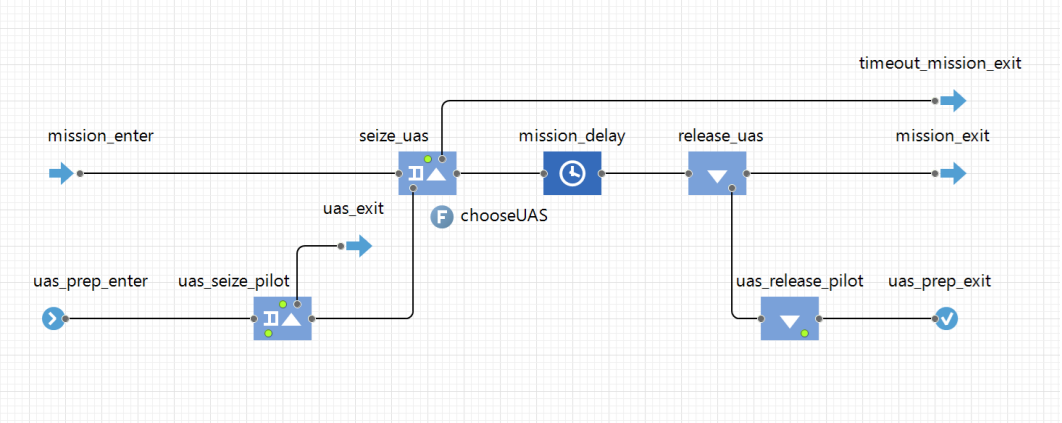


FIGURE 4.6: Discrete event simulation blocks defining the mission process flow through the main program of the simulation.

The queue within the *seize_uas* block is a first in first out (FIFO) queue with priority enabled. This means the mission agents will be queued in order of generation unless they have a higher priority value, in which case they will enter the queue in order of priority. The UAS allocation is determined through the *chooseUAS* function. This takes the pool of UASs (*i.e.* all that are in the service) and for each (if it is idle) the function calculates if the UAS is suitable in terms of range and endurance to complete the mission taking into account the current weather conditions. If the mission task has an unknown or variable duration then a minimum-time-on-task value is added to the calculation to ensure part of the flight is spent doing the useful element. The minimum-time-on-task value is set as a policy on the mission agent via its mission generator which, in turn, is set via the mission details. It can be set as either a value with units of time or a percentage of the variable duration. Once a list of suitable UAS platforms is found for the mission agent, the nearest UAS platform to the mission location is selected from the list to complete the mission[4]. The *chooseUAS* function is also called each time a UAS becomes idle to ensure all suitable UAS platforms are considered for the mission agents in the queue.

[4]During analysis of the results obtained in the case study described in Chapter 5, it was found that this method had an error when the list of suitable UASs were all stationed at the same operating base. This meant that they all were the same distance away from the mission location and therefore the first in the list was selected each time. To resolve this, a further policy can be implemented which states which platform type is preferred when this scenario is presented. This policy could be attached to the mission agent in a similar way as the minimum-time-on-task policy described above.

The selected UAS platform is then prepared for the mission. This follows the path from the lower left of Figure 4.6 starting at *uas_prep_enter* and ending in the *seize_uas* block. During the preparation, the UAS unit seizes a pilot[5] from the operating base it is associated with to oversee the mission, or it waits for a pilot to become available. Note that this queue also has a shelf-life similar to the mission agent queue described above and will trigger a mission time-out should a pilot not become available in the time limit. This study has designed the framework such that for a flight to occur a pilot is required for the duration of the mission (in whatever capacity the service requires *i.e.* full manual control, or acting purely as an observer or safety pilot). This design attribute is not intended to directly determine the level of autonomy by which the UAS performs the mission.

Once the mission agent is allocated a UAS and the required resources, it proceeds to the *mission_delay* block in Figure 4.6. This is the *Activate Mission* block in Figure 4.2 on Page 64. The mission delay is controlled by the UAS agents state-chart and only once the UAS has completed the mission's task and returned to an idle state is the *mission_delay* block exited. The UAS state-chart is described and discussed in Section 4.3.2. The basics of the state-chart are shown in the asset actions column of Figure 4.2. On exit from the *mission_delay* block, the allocated UAS and required resources are released via the *release_uas* and *uas_release_pilot* blocks respectively, and made available for the next mission agent in the *seize_uas* and *uas_seize_pilot* queues. All the relevant data collected during the mission is stored for the end of the simulation. This completes the mission process flow.

Multiple mission agents can be in the process at any one time and they can represent different mission types from different mission generators. The mission agent contains all the information required to describe the mission and the task.

4.3.2 UAS agent

The UAS agent contains all the parameters required to define the platform in terms of performance, reliability and maintenance. It also collects all the data associated with its usage throughout the service. The main component of the UAS agent is the state-chart. A screenshot of the state-chart taken from the graphical editor in AnyLogic is shown in Figure 4.7.

Figure 4.7 can be read by starting from the entrance at the top and following the transitions between states in the direction of the arrows. The green coloured state blocks indicate those which occur at the operating base. Whereas, the blue coloured state blocks are when the UAS is airborne.

[5]Here, a *pilot* does need to be an individual as discussed in Section 3.2.3 and can represent an entire crew if required.

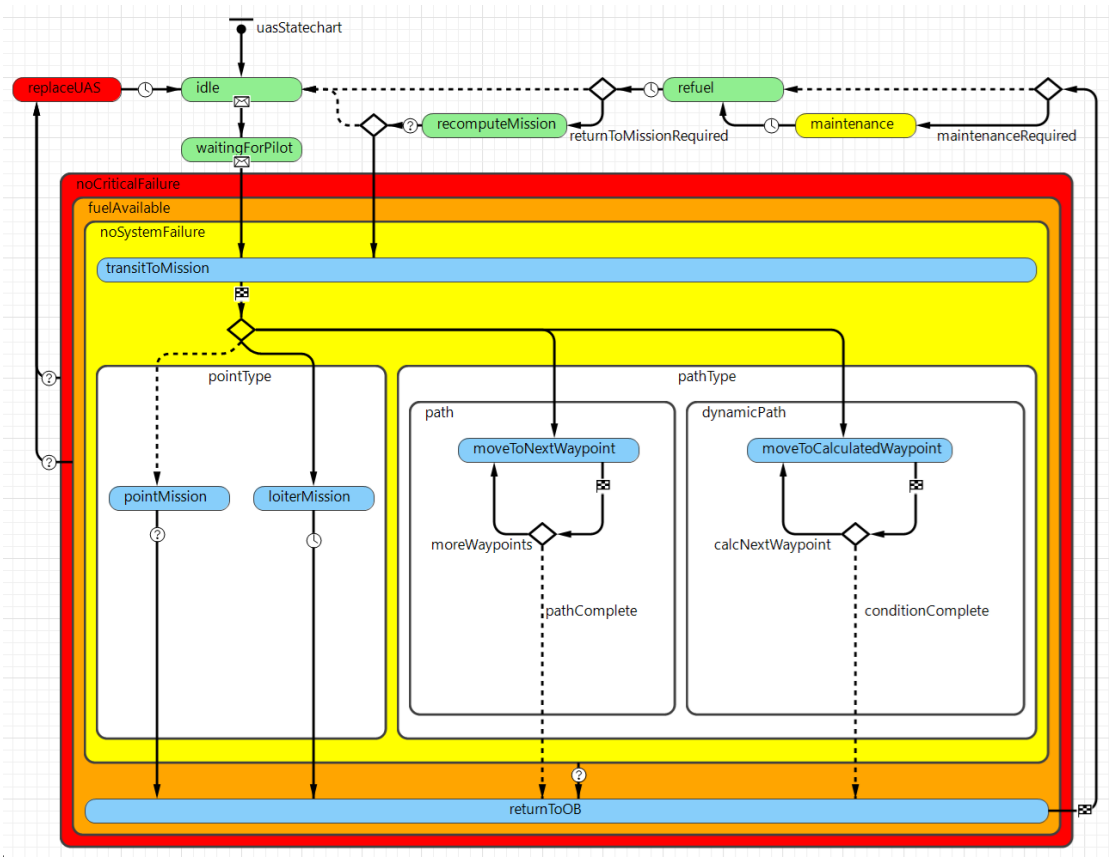


FIGURE 4.7: The UAS state-chart used to process the actions of the UAS to complete the assigned mission.

The UAS agent is initialised in the *idle* state and waits to be allocated for a mission. On allocation, the state is changed to the *waitingForPilot* state as described in the main discrete event model above. Once assigned a pilot, the UAS transits to the mission location. It can be seen in Figure 4.7 that all the airborne states in the state-chart are encapsulated in the *noCriticalFailure* (red block) and *fuelAvailable* (orange block) states. Both of these states are exited based on a change in condition which can be triggered by an event or as a consequence of a calculation, resulting in the UAS being lost and requiring replacement. This route can be seen in Figure 4.7 by following the left hand side of the state-chart where the transition arrows lead to the *replaceUAS* state block. As mentioned in Section 3.2.4 the length of time to complete the UAS replacement can be set per UAS type and defines the time to transition between the *replaceUAS* and *idle* state blocks.

The transit to mission location and the state blocks for performing the mission's task are encapsulated by a *noSystemFailure* (yellow block) state. When the condition to trigger the change in this state is met, the UAS is transitioned into the *returnToOB* state (*i.e.* return to operating base) where a maintenance operation is performed.

The failure states are triggered once the time-since-last-failure is equal to greater than the sampled value from the failure-time distribution (as discussed in Section 3.2.4). Due

to the event-based nature of this simulation, these conditions are checked before every event that increases the flight time. This is because the flight time is not increased at regular time intervals, but just on events. This allows an failure-event to be scheduled if the failure condition is due to occur within the event's time step.

On arrival at the mission location the state-chart reaches a branch (symbolised by a diamond in Figure 4.7). The route that is followed depends on the type of task that has been assigned to the UAS. Note that the white blocks in the state-chart do not serve any other function other than indicate the type of task (*i.e.* they act as a visual aid when running the simulation in animated mode). The blue state blocks within the white blocks, for example *loiterMission*, contain the calculations and logic required for the UAS to perform that particular task.

Once the task is completed or the UAS cannot continue the task any more due to running low on energy, the UAS returns to the operating base. If maintenance is required due to system failure or preventative maintenance scheduling then this is performed. The UAS is then refuelled and the amount of consumables used are recorded. If the task is still not complete to a satisfactory level (which can be set in the mission details), the UAS will recompute the mission and continue the operation by flowing through the states as described above. Finally, if the mission is complete the UAS is returned to an *idle* state and the mission delay in the main discrete event model is exited.

Each state that moves the UAS calculates the range (and endurance) possible using the given performance values and the state of energy reserve (*i.e.* fuel or battery capacity remaining). This range is compare to the range required to complete the next step of the mission plus the range required to return to its operating base while taking into account the current wind conditions and any fuel reserve policies. These calculations are also completed each time new wind conditions are generated. If at any point the UAS does not have enough fuel available to return to its operating base then the *fuelAvailable* state is exited and the UAS is classed as a total loss and requires replacement. This condition should only be reached due to an adverse change in wind conditions.

The wind conditions are communicated to the UAS agent when the weather event generator triggers a change in conditions. The wind details are then used to adjust the ground speed of the UAS and the aerial distance to fly to its destination based on the resultant vector between the wind and UAS's current flight regime (*i.e* airspeed and altitude). The wind speed value is sent as a measurement at 10 m above ground level so that it can be adjusted to take into account the wind profile with altitude.

The payload module is linked to the UAS agent as shown in the model hierarchy diagram presented in Figure 4.3 on page 64. The payload's parameters are passed to the UAS so they can be used to calculate the platform's total mass and, if required, the values to calculate the dynamic path waypoints.

In the case where an EO sensor is used to calculate the dynamic path, the average ground footprint width and path length are used to determine the area scanned. If the task type requires the payload to trigger the stop-task condition (*e.g* detecting a target according to the Johnson criteria) then this condition is assessed at the beginning of each path by check whether the target is in the polygon drawn by the sensor's field of view. If it is, then the path length is recalculated to be the distance at which the target is detected.

Chapter 5

Results and analysis

This chapter presents the findings from running the simulation of the model detailed in the earlier chapters. The model was set up to run the case study described in Section 5.1 which considers the design of a service to support coastguard operations on the south-coast of the UK. The results of the case study are presented and discussed in Section 5.2 with focus on the performance of the model and simulation. Finally, the findings from this case study are summarised in Section 5.4.

5.1 Case study

The case study used to exercise the model and obtain results is based on supporting coastguard and Search and Rescue operations on the south-coast of the UK. To build up the definition of the case-study service, some research was completed on these operations. Recently, the Maritime Coastguard Agency in the UK opened a Request for Proposal to provide HM Coastguard search and rescue helicopters, planes and remotely piloted drones under the UK Second Generation Search and Rescue Aviation programme (known as UKSAR2G) [14]. One of the key innovative solutions under consideration in this RFP is the use of remotely piloted drones.

The data provided by the UK government on search and rescue helicopter statistics highlights the usage of the SAR helicopters for their multitude of operations. A snippet of this data is plotted in Figure 5.1 which captures a year of helicopter tasks[1] around the south-coast of the UK. When plotted, this data set clearly shows the spread of task locations.

The data set also contains additional metadata which can be useful for fine tuning the service details. The task count displayed in Figure 5.1 equals 226 with 129 of them under

[1]It should be noted that the data set plotted spans between 01-04-2020 and 31-03-2021. Therefore, the figures recorded in April 2020 captures the impact on SAR helicopter tasks of national lockdown in response to COVID-19.

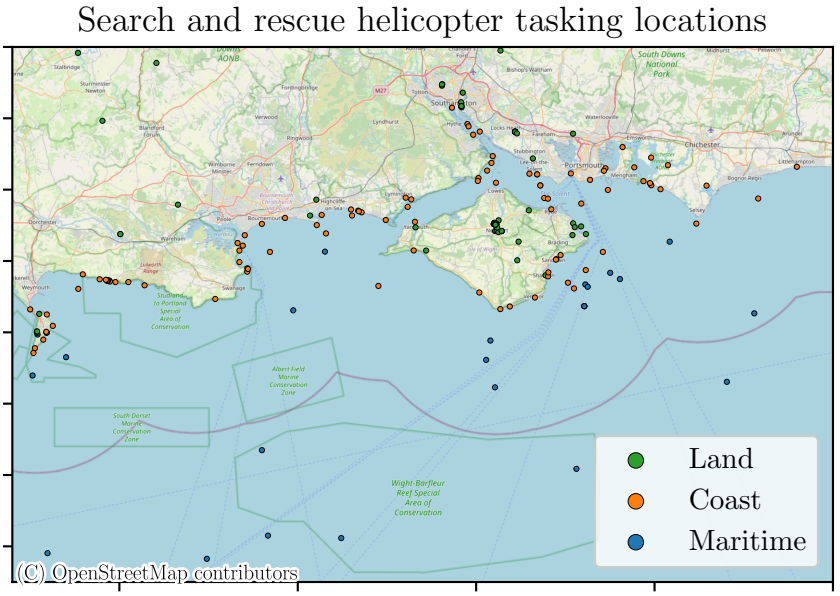


FIGURE 5.1: Search and rescue helicopter task locations between 01-04-2020 and 31-03-2021 in the UK over the south-coast. This covers all types of task and task outcomes and is categories on the map by task location. Data available from `www.gov.uk/government/statistical-data-sets/search-and-rescue-helicopter-sarh01`.

the *coast* and *maritime* categories. For the purpose of this case study, categorising the tasks by type (as opposed to location) was also a useful insight. This can be seen in Figure 5.2. The type of tasks covered were pre-arranged transfer, rescue/recovery, search only and support. Therefore, by removing pre-arranged transfer from the total count of 226 the remaining types totalled 168 (as this is not a task the drone is expected to fulfil).

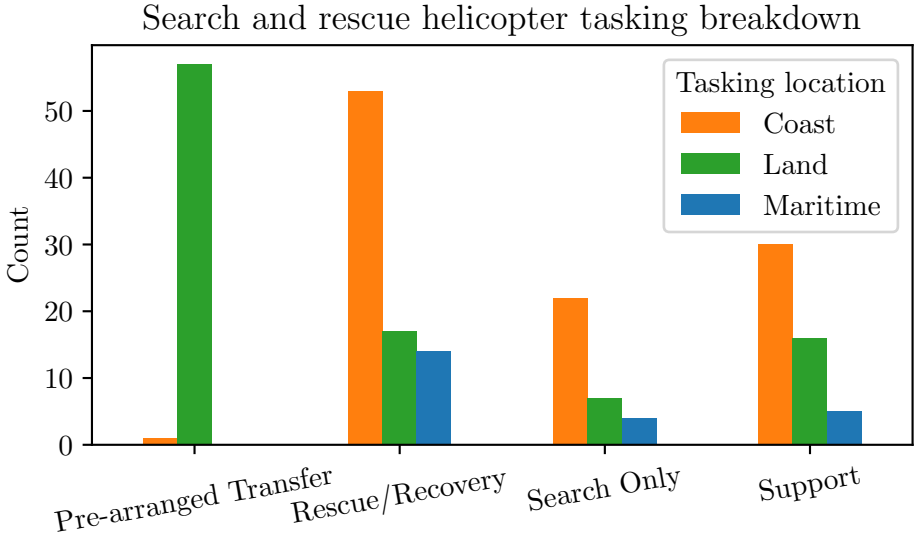


FIGURE 5.2: Search and rescue helicopter tasking breakdown between 01-04-2020 and 31-03-2021 in the UK over the south-coast. Data available from `www.gov.uk/government/statistical-data-sets/search-and-rescue-helicopter-sarh01`.

Another data set studied to build up the definition of the service was that of the Royal National Lifeboat Institution (RNLI) [136]. This data set contained the locations of incidents attended by a lifesaving craft. A snippet of this data is plotted in Figure 5.3 which shows the incidents that occurred around the south-coast of the UK over one year. This plot shows a total of 829 launches served by 16 lifeboat stations across the map. It can be seen that the incidents tend to be concentrated in groups around the coastline matching up to areas of higher population density, water-sport areas and tourist hot-spots. This is also typically where the RNLI have a lifeboat station.

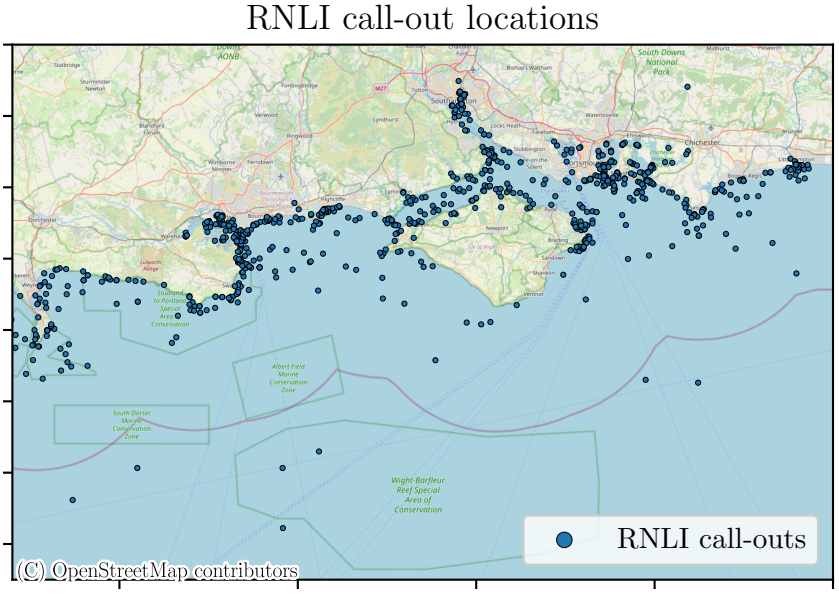


FIGURE 5.3: Royal National Lifeboat Institution (RNLI) lifeboat call-outs between 01-01-2020 and 31-12-2020 where a lifesaving craft has gone to the aid of persons in need. Data available from `rnli.org/about-us/our-research/rnli-open-data`.

The combination of these two data sets was used in the formation of this case study's inputs. The details of which are described in Section 5.1.1. The locations and frequency of the historic incidents were used to form multiple probability distribution functions that represented the data sets, but introduced a stochastic element to the simulation runs.

The aim of the case study is to ascertain which operating bases to use and what combination of UASs will create the optimum service based on the value function.

5.1.1 Service Details

This section is split into sub-sections to build up the picture of the service.

5.1.1.1 Mission generators

After studying the data presented in the two data sets described in Section 5.1, a selection of mission generators were created. Firstly, a regional polygon-based mission generator was introduced which covered the entire south-coast region (using an exclude-polygon over the Isle of Wight). This was used to represent the SAR helicopter data set and produces random points within the polygon at a defined frequency distribution to match the data set's frequency.

Secondly, a further 10 arc-based mission generators were created in an attempt to reproduce the localised incident hot-spots seen in the RNLI data set (see Figure 5.3). These were based at 10 locations across the south coast and had radial-distance distributions that followed a triangular distribution, each of which were individually specified. These also each had an individually defined frequency distribution that was set to mimic the data set frequencies.

The result of all 11 mission generators can be seen in the snapshot of one simulation run over a 6 month time period shown in Figure 5.4. It is worth noting that the locations and frequencies are created from probability distributions and therefore when the model's random number generation is from a random seed the locations and count will vary between repetitions. This is one source of the stochastic nature of the model.

The total count of SAR missions during this random seeded 6 month time period was 441. The total number of tasks and call-outs recorded by combining the supporting data sets (and dividing to match the time period[2]) was 499. Therefore the frequency distribution seems to be acceptably matched.

Finally, one daily mission generator was created to represent a task of pollution monitoring of the Solent region. This was added to model a component of the UKSAR2G commercial strategy, focusing on Lot Three which includes surveillance and pollution response [14]. This task was set to be a 40 minute duration actioned at 10:00 every day taking place in the Solent region between the Isle of Wight and Portsmouth.

Each mission generator was provided with a *mission value* used to represent the stakeholders' view of the importance of the mission. This was set highest (400) for the south-coast region missions, equal (100) for all the RNLI style missions and lowest (75) for the daily task. In turn, this value was used to prioritise the missions and, ultimately, provide a *mission score* which represented how successful the UAS was at completing that mission.

Also, each mission generator was provided with a *shelf life* for the missions created. This value was used to determine how long the mission would stay unassigned before it was cancelled and given a score of zero. This can be thought of as the notice to respond time.

[2]This took the total of the SAR helicopter data set with the type *pre-arranged transfer* removed and the total of the RNLI call-outs from the year period and divided by two to give an estimate of the total expected in a 6 month period.

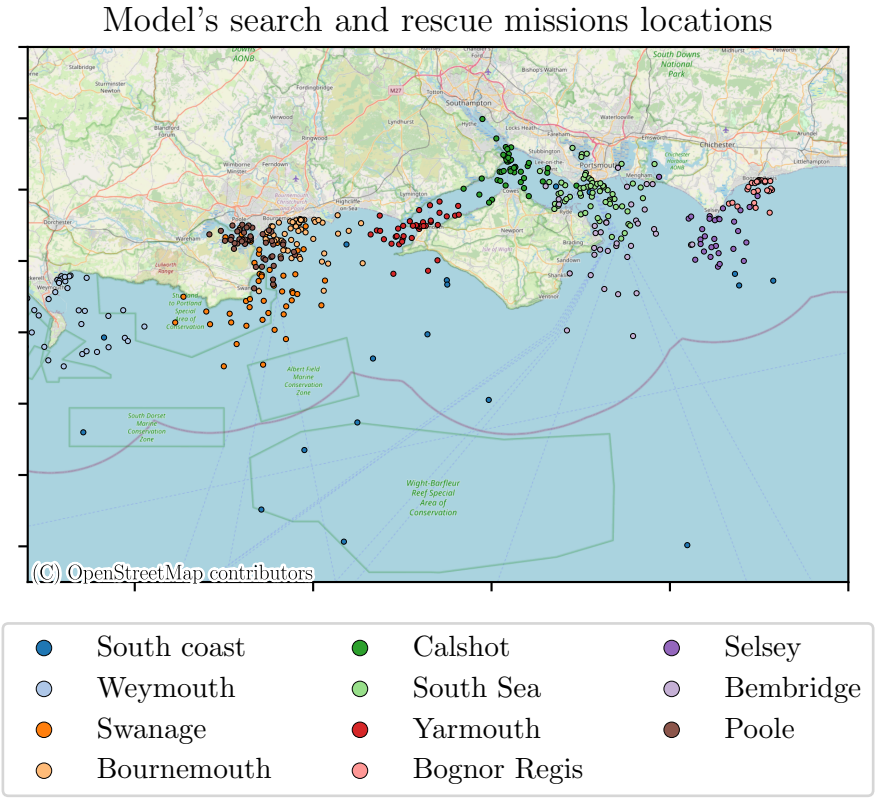


FIGURE 5.4: The model of the case study's search and rescue mission locations categorised by the source mission-generator. This is a snapshot of one random-seeded simulation run and shows a total of 441 missions.

This was set highest, at 120 minutes, for the daily task due to its low urgency, equal at 30 minutes for the RNLI style missions to represent a short notice to respond and finally 60 minutes for the south-coast region missions to represent a longer notice to respond.

5.1.1.2 Operating bases

As this case study was using the data set of RNLI call-outs it was decided to use RNLI lifeboat stations as the operating bases, or if available, a nearby airfield (to ensure the correct facilities are in place). Most lifeboat stations have an area of grass or tarmac large enough to facilitate the launch and recovery of the UAS considered in this case study.

The following three locations were selected as the candidate operating bases. These can be seen geographically located in Figure 5.5.

1. Lymington RNLI lifeboat station,
2. Swanage RNLI lifeboat station,
3. Bembridge Airport (near to Bembridge RNLI lifeboat station).

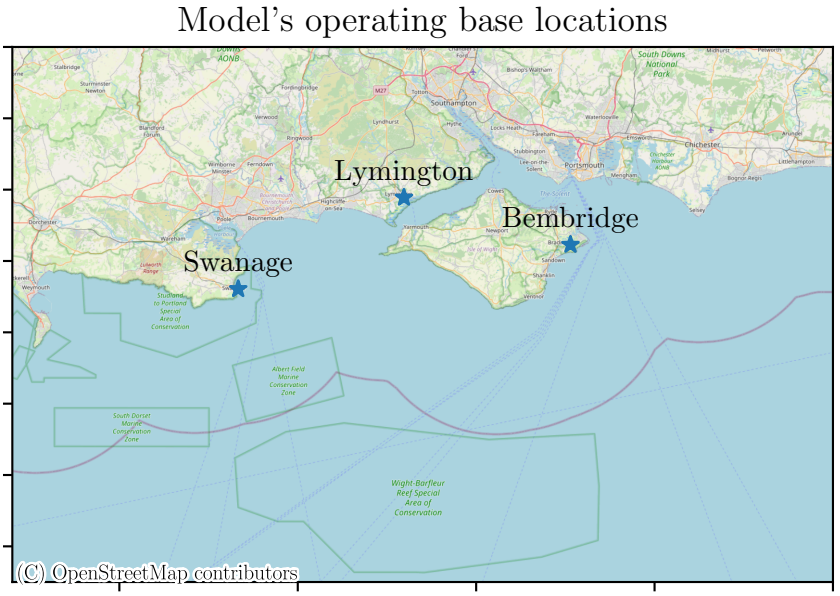


FIGURE 5.5: The operating base locations used in the case study. They are are also assigned identifying numbers such that Lymington is 1, Swanage is 2 and Bembridge is 3.

These three locations were selected as they provide a good coverage of the entire south coast region considered in this case study. To account for facility overheads, each operating base was assigned a daily cost. This was set to be different at each location and is detailed in Table 5.1.

TABLE 5.1: Operating bases' daily costs.

	Daily cost (£ / day)
Lymington	80
Swanage	60
Bembridge	50

The distribution of the UAS platforms to the operating bases for the different candidate designs was as follows. Each operating base was assigned from zero up to a maximum of four UAS platforms, comprising of a maximum of two each of two different platform types. More details on the UAS platform types selected is provided in Section 5.1.2.

The total permutations with repetition, ${}^{n}P_{r}$, is given by the formula

$$ {}^{n}P_{r} = n^{r} \tag{5.1} $$

where n is the number of possibilities (in this case $n = 3$ *i.e.* 0, 1, or 2 units) and r is the number of choices (in this case $r = 2$ *i.e.* UAS 1 and UAS 2). Therefore, the total number of permutations per operating base is 9. These permutations can be seen in Table 5.2.

TABLE 5.2: Individual operating base UAS permutations for the two assigned UAS.

	#1	#2	#3	#4	#5	#6	#7	#8	#9
UAS 1	0	1	2	0	1	2	0	1	2
UAS 2	0	0	0	1	1	1	2	2	2
Total	0	1	2	1	2	3	2	3	4

Now, using Equation 5.1 again, the total number of permutations with repetition using the three operating bases can be found. Here, $n = 9$ (*i.e.* the number of permutations per operating base) and $r = 3$ (*i.e.* the number of operating bases), therefore this results in ${}^{n}P_{r} = 729$. Or, put differently, 729 candidate designs.

Additional, was an investigation into restricting the number of pilots available per base on a global level. This was achieved by setting the maximum number of pilots per operating base to either 1 or 2. This followed the rule that if there are zero UASs at an operating base then there will be zero pilots (*i.e.* the operating base is not used and therefore incurs no daily costs nor have any personnel assigned to it). Then, the number of pilots at each operating base will increase with the number of UAS units (at that particular operating base) up to this set value. This increased the number of candidate designs by a factor of 2 such that there were 1458.

5.1.2 UAS agents

This case study looked at the combination of two UAS platforms. The platforms were selected from the pool of UAVs designed and manufactured at the University of Southampton due to the Author's in-depth knowledge of the platforms and access to performance and cost data. The two platforms were selected such that they were varied in performance, design and cost.

UAS 1 - Spotter

Spotter (which stands for Southampton Platform for Observation, Tracking, Telecommunications and Environmental Reconnaissance) is a twin engine, twin boom monoplane. The two engines are single cylinder, four-stroke petrol engines mounted in a tractor configuration (see Figure 5.6). The design of Spotter focuses on component redundancy and flight safety by doubling up all control surfaces and using two power plants to provide thrust and generate power for two power buses. The payload is mounted to a pylon below the fuel tank near the centre of gravity (CoG) to allow for a versatile range of payload options. Also, by having both the payload and the fuel tank on or near the CoG, it allows the aircraft to fly at its maximum take-off mass or at its empty mass with minimal impact on the flight characteristics. It has a maximum take off mass of 35 kg and and an empty mass of 24 kg which allows for 6 kg of fuel and 5 kg of payload.

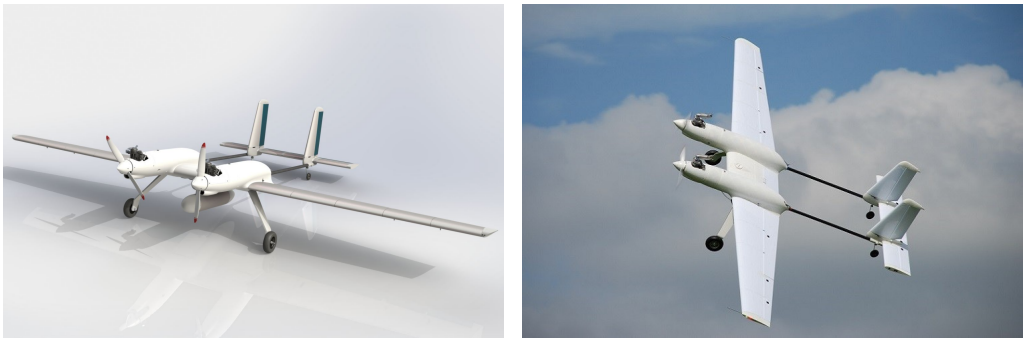


FIGURE 5.6: UAS 1 - Spotter

The design was developed through an iterative process through research projects at the University of Southampton. The initial design philosophies of Spotter started in the DECODE project[3] which looked at system level trade-offs. After the DECODE project, the research team developed the 2SEAS-20 platform as part of the 2SEAS-3i project[4] which investigated the use of UAVs as a service for coastal zone management. Spotter was conceived through the iterative development of this platform. More information about the design and design process using value driven design and additive manufacturing techniques can be found in Ferraro's thesis [42].

UAS 2 - Valerie

Valerie (which stands for Vertical Ascent and Landing for Enhanced Research, Innovation and Exploration) is an all electric power-plant quad-plane design utilising a flying wing and a pusher configuration capable of vertical take-off and landing (VTOL) (see Figure 5.7). The design of Valerie stems from the CASCADE (Complex Autonomous Aircraft Systems Configuration, Analysis and Design Exploratory) programme[5]. The platform was designed under the Open Aircraft Project which is an initiative to create small UAS designs which meet the requirements of the science community when using fixed-wing drones in challenging remote locations [48]. On the completion of the design and after further thorough testing the design will be made freely available to all.

The design requirements of the platform were primarily driven by the mission requirements for volcanic ash sampling [137] over Volcán de Fuego in Guatemala. Based on previous flight profiles, this demanded a service ceiling of 5000 m, an endurance of 30 minutes at altitude (or 60 minutes at sea level) along with a minimum cruise speed of 20 ms^{-1}. It also had to be capable of carrying a payload of 0.8 kg or greater.

The latest iteration of the Valerie platform produced from these requirements, has a maximum take-off mass of 11.5 kg and an empty (structural) mass of 6.9 kg. It

[3]The DECODE (Decision Environment for COmplex DEsigns) project concluded in 2012 and investigated design decision making tools and processes in the context of system design. For more information of the DECODE project go to `www.southampton.ac.uk/~decode/`.

[4]For more information on the 2SEAS-3i project visit `www.2seas-uav.com`.

[5]For more information on the CASCADE programme please visit `www.cascadeuav.com`

accommodates 0.9 kg of VTOL batteries and 2.6 kg of cruise batteries leaving 1.1 kg for the payload. More information on the configuration choice rational and design methodology can be found in [48].

FIGURE 5.7: UAS 2 - Valerie

In comparison, UAS 1 is the more expensive platform in terms of unit cost, cost per flight hour and consumable costs. However, it is also the more capable platform with a higher range, endurance and payload carrying capability. Also, UAS 1's design philosophy regarding component redundancy makes the platform more reliable.

5.1.3 Weather model

The weather model used in this case study consisted of temporal wind speed and direction data. This was collated from historical data records for Southampton and applied globally to the simulation. For the wind direction, a custom distribution was created in the model for each month that followed the historical data set. Based on the month of the simulation clock, the corresponding distribution was sampled every 6 hours to set the wind direction. The distributions can be seen in Figure 5.8.

It can be seen in Figure 5.8 that the prevailing wind direction is from the south-west with slight variations between each month. Note that the radial axis denotes the percentage of time spent with the wind coming from that direction (*i.e.* this is not a wind rose plot and the radial values do not indicate wind speed). The summer months show less variation in wind direction compared to the other months, with March being the most variable.

For the wind speed, the monthly average wind speed at 10 m above ground level was used to create the wind speed distribution. The monthly averages are plotted in Figure 5.9. The distribution was then formed by using the Rayleigh distribution function. The sample generated was then used to find the wind speed at altitude using the log wind profile. Both of these distributions are shown for January in Figure 5.10. Therefore, the weather model was another stochastic input to the model that has an influence on the output of the simulation.

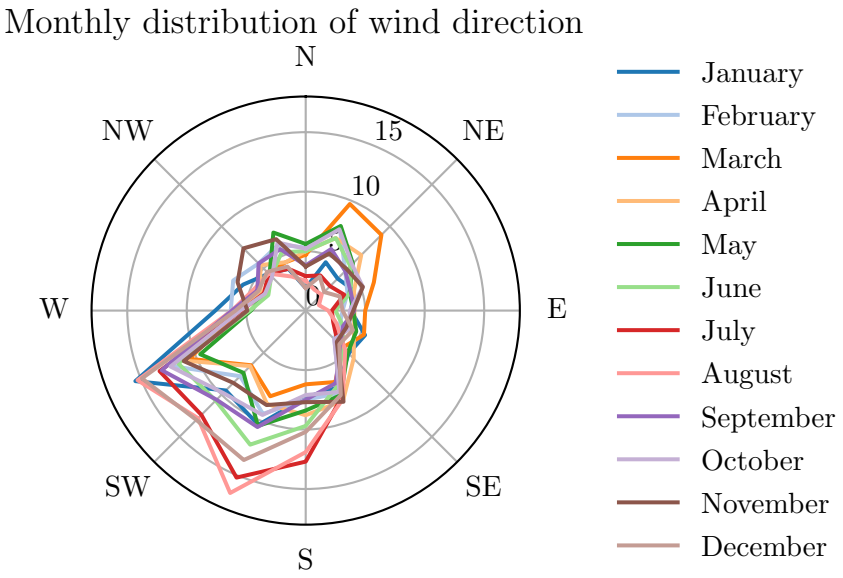


FIGURE 5.8: The monthly distribution of the wind direction in Southampton used in the case study. The radial-axis denotes the percentage of time spent with the wind coming from that direction. From this plot it can be seen that the prevailing wind direction is from the SW.

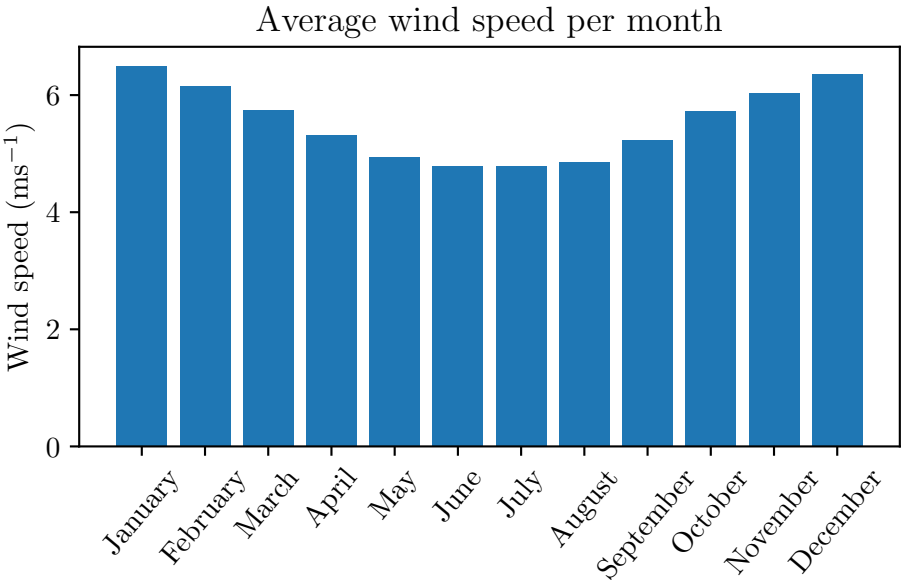


FIGURE 5.9: The average wind speed at 10 m above the ground in Southampton per month used in the case study.

5.1.4 Value model

To demonstrate how the choice of value model can lead to different optimal designs, two different value models are studied. Due to the challenge of monetising the mission success score (*i.e.* the service effectiveness of the design), a cost-benefit analysis approach was

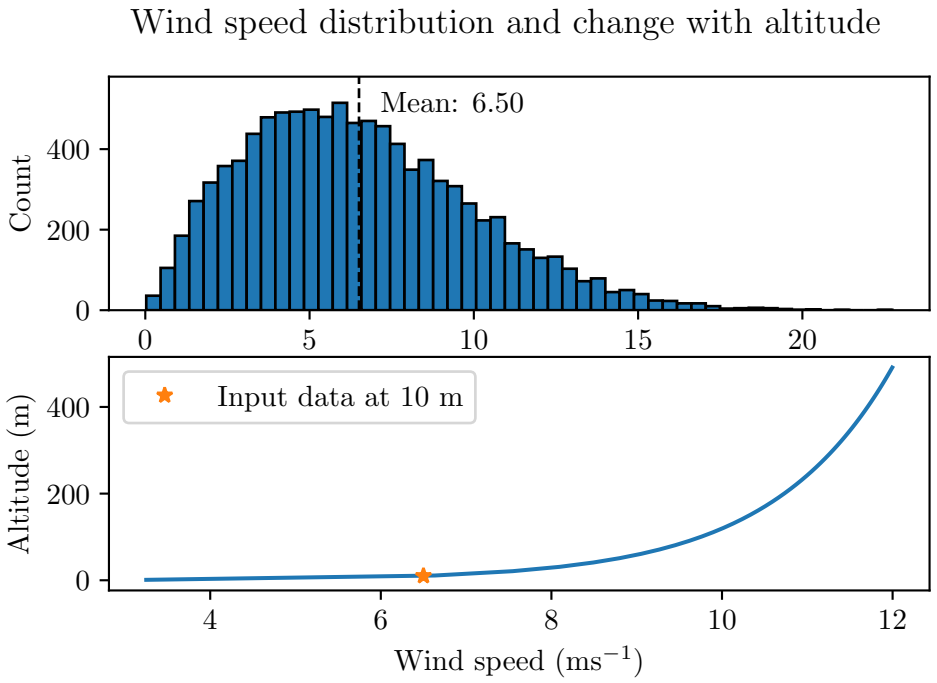


FIGURE 5.10: Demonstration of the wind speed distribution and the log wind profile showing change with altitude (using January's mean wind speed of 6.50 ms^{-1}).

taken. This allows the user of the model to compare the level of effectiveness against the associated costs on a bi-dimensional graph.

The two values models considered in this case study looked at firstly shifting the emphasis towards maximising the mission success score and secondly to reducing the total service cost. The value function defines the gradient along which the value increases and therefore can be plotted on a bi-dimensional graph (of mission success-score against cost) to demonstrate the value mapping.

The shift in emphasis is achieved by adjusting the weighting on the parameters that form the benefit component of the model following Equation (2.24). As the case study only considers a time period of six months the discount rate has been neglected in the cost computation.

5.2 Results and discussion

The following sections discuss the results obtained by running the simulation of the case study described above.

5.2.1 Convergence study

The simulation produces a vast amount of raw data regarding the service performance. This includes, for example, the total mission success score, the overall cost (and the cost breakdown), the total flight time of each UAS in the simulation, the amount of fuel and electricity consumed. These outputs are used in the value model, but also remain accessible for the user to analyse the results in more detail. This allows the reasoning for the optimal design to be transparent and confirm the effect of any assumptions.

The service simulation is influenced by several variables that have randomly generated inputs (for example, the mission frequency, locations and task period, and the modelled weather conditions all take input values from probability distributions). Therefore, due to the stochastic nature of the service simulation, each replication of the simulation for a particular design candidate will produce different results. The Monte Carlo method is best suited to this situation and was therefore used to obtain the results from the simulation. By increasing the number of replications, the sample means of the design candidates should approach the real means with increasing accuracy. However, there is a compromise between accuracy and computational cost.

Both the confidence interval method and the graphical study method highlighted in Robinson [82] were used to assess the number of replications required. The confidence interval method is a statistical method for showing how accurately the mean is being estimated. It is based on monitoring the ratio of the confidence interval to the sample mean as a function of the number of replications. Here, the confidence interval, CI, is calculated as

$$CI = \bar{X} \pm t_{n-1,\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \tag{5.2}$$

where $\bar{X}$ is the cumulative mean, n is the number of replications, and $t_{n-1,\frac{\alpha}{2}}$ is the value from Student's t-distribution with $n-1$ degrees of freedom and a significance level of $\frac{\alpha}{2}$. σ is the standard deviation of the output data from the replications and is defined as

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)^2}{n-1}} \tag{5.3}$$

where X_i is the result from the ith replication. The significance level $\alpha = 5\%$ was used to give a 95% confidence interval.

Figures 5.11 and 5.12 show the results of the confidence interval method and a graphical study of the cumulative mean for the mission-success score versus cost outputs respectively for four randomly selected design candidates. The simulation was run for 850 iterations

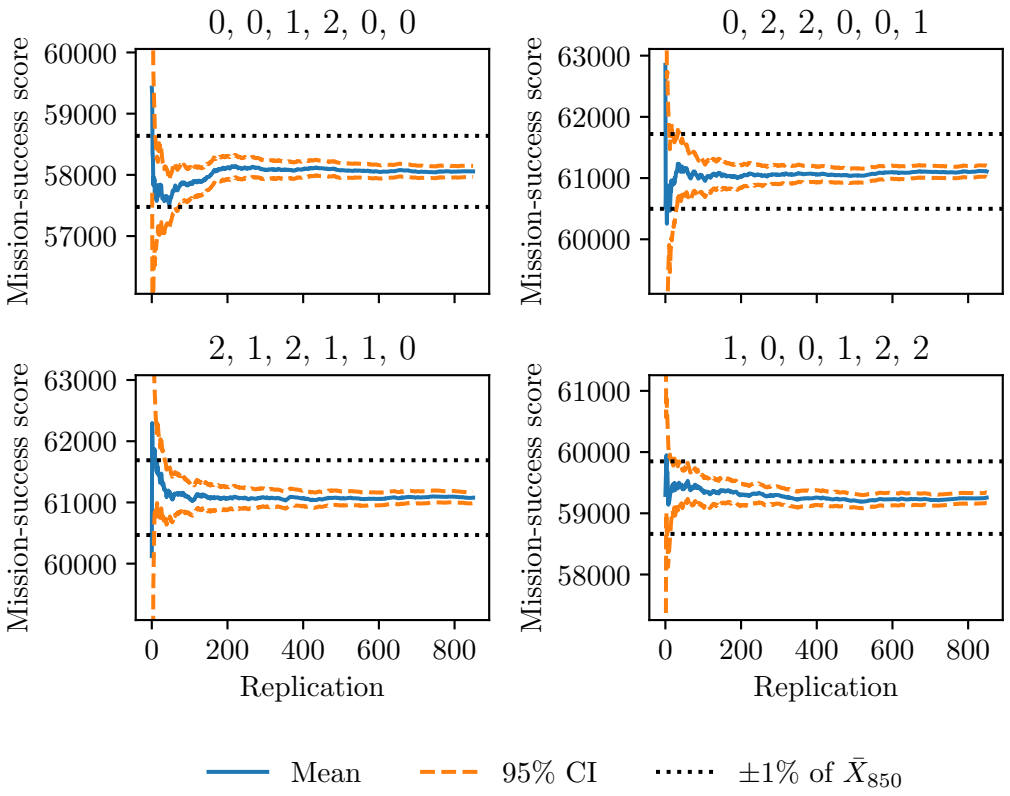


FIGURE 5.11: Graphical study of the cumulative mean and 95% confidence interval (CI) of the mission-success score for four randomly selected design candidates. The titles of the plots refer to the distribution of platforms: the first three numbers relate to UAS 1 and signify the distribution to Lymington, Swanage and Bembridge respectively; the last three are with respect to UAS 2.

and the graphs show the 95% confidence interval and ± 1% with respect to the asymptotic value of the cumulative mean.

The titles of the subplots in the figures introduce the candidate design naming scheme. They refer to the distribution of platforms at the operating bases: the first three numbers relate to UAS 1 and signify the distribution to Lymington, Swanage, and Bembridge respectively; the last three numbers are with respect to UAS 2. For example, 0, 0, 1, 2, 0, 0 in the top left plot in Figure 5.12 signifies one UAS 1 at Bembridge and two UAS 2s at Lymington.

The graphical method proposed by Robinson [82] uses the plots shown in Figure 5.11 and 5.12. As more replications are performed the plot of the cumulative mean should become a flat line, with minimal variability and no upward or downward trend. The number of replications required is defined as the point where the line becomes flat, and performing further replications will only give marginal improvements to the mean value [82].

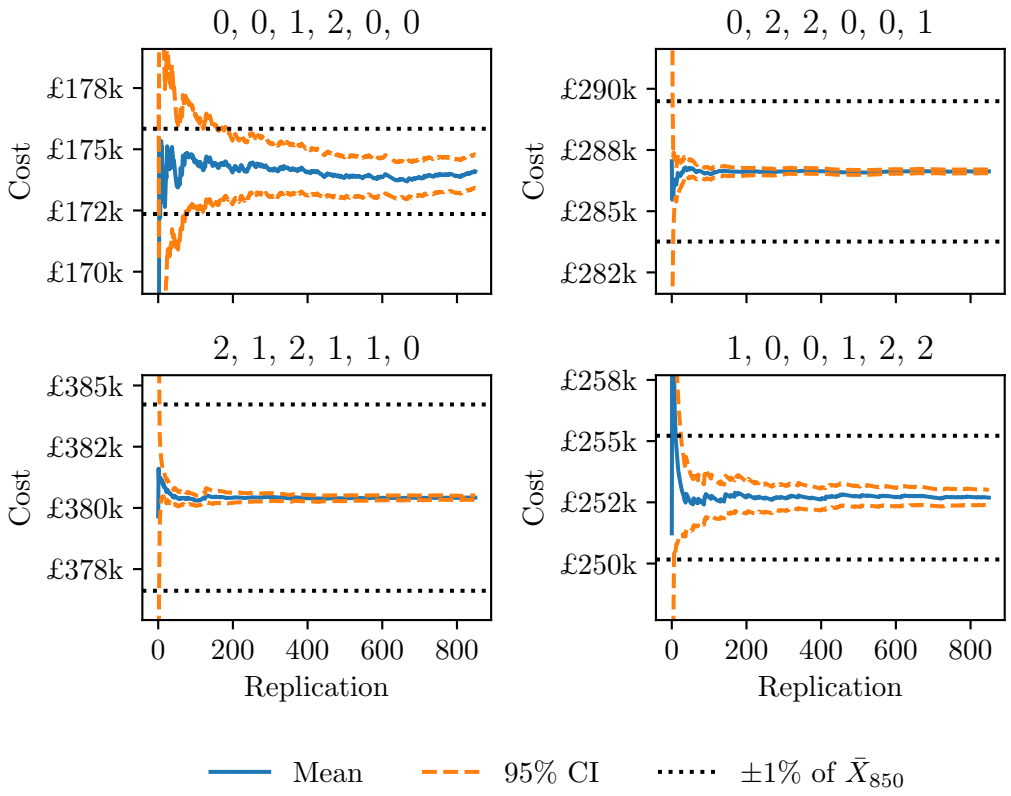


FIGURE 5.12: Graphical study of the cumulative mean and 95% confidence interval (CI) of the total cost for four randomly selected design candidates. The titles of the plots refer to the distribution of platforms: the first three numbers relate to UAS 1 and signify the distribution to Lymington, Swanage and Bembridge respectively; the last three are with respect to UAS 2.

On analysis of Figure 5.11 (number of replications against mission-success score), it can be seen that around the 100 replications mark all the 95% confidence intervals are within the ± 1% error of the converged mean and therefore it should be sufficient to stop the replications there. However, the plot of the cumulative mean has not reached a sufficiently flat profile. It can be seen still rising in the top left subplot at around 100 replications. Therefore, based on Figure 5.11, the number of replications should be set at 250.

By applying the same analysis technique on the plots shown in Figure 5.12 (number of replications versus service cost), it can be seen that the top left plot stands out as being significantly different to the others. These show a sufficiently flat cumulative mean after 250 repetitions and the 95% confidence interval is within the ± 1% error within a very small number of replications. The top left plot of Figure 5.12, on the other hand, shows that the 95% confidence interval width is just within the ± 1% error bounds by 200 replications, but the cumulative mean has not fully settled to a flat line. Due to this outlier further analysis was completed on the full set of design candidates after 250 replications to ensure the cumulative means had acceptable levels of accuracy.

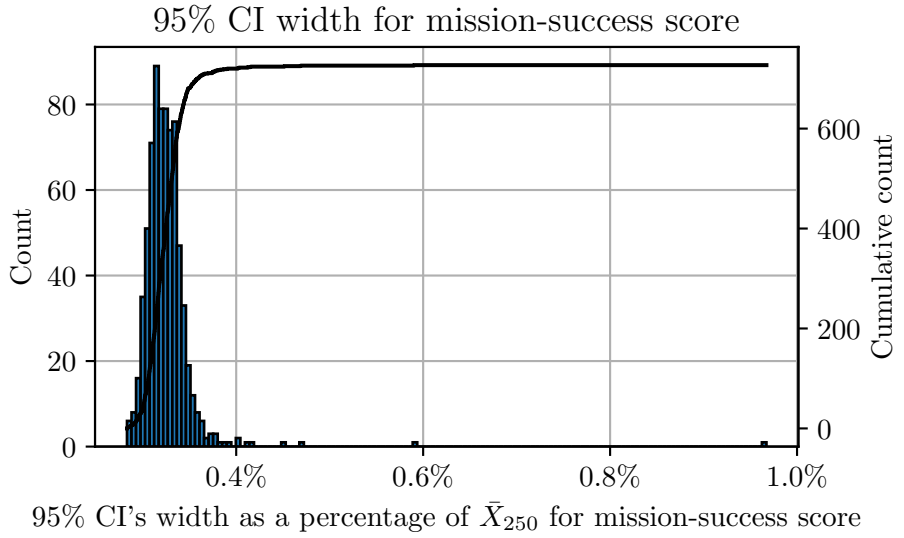


FIGURE 5.13: Histogram and cumulative distribution plot of the mission-success score confidence interval width as a percentage of $\bar{X}_{250}$ for the 728 design candidates after 250 replications (excluding the zero UAS 1 and UAS 2 candidate due to it skewing the histogram).

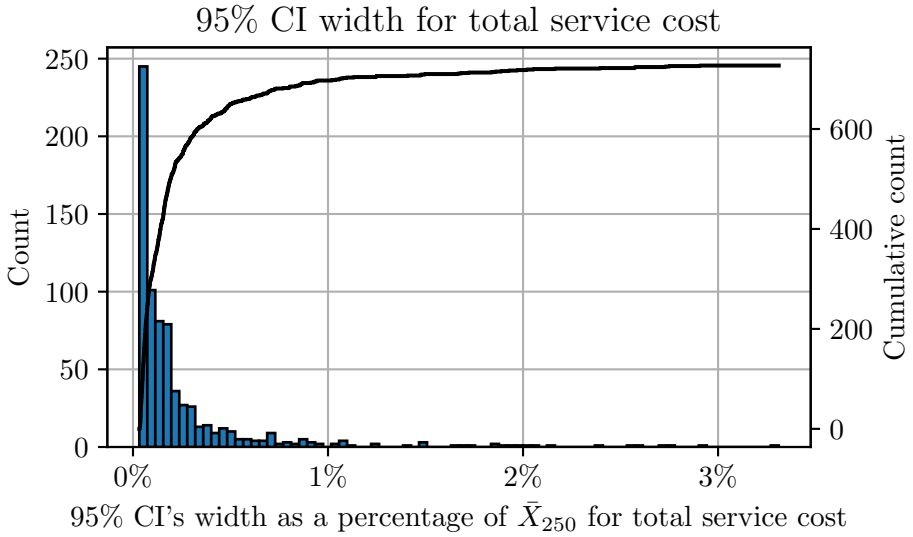


FIGURE 5.14: Histogram and cumulative distribution plot of the total cost confidence interval width as a percentage of $\bar{X}_{250}$ for the 728 design candidates after 250 replications (excluding the zero UAS 1 and UAS 2 candidate due to it skewing the histogram).

Once all design candidates were simulated with 250 replications, histograms were produced of the 95% confidence interval width as a percentage of $\bar{X}_{250}$ of two important outputs: the mission-success score and the total cost. These histograms are shown in Figure 5.13 and Figure 5.14. The number of bins in the histograms was chosen using the Freedman-Diaconis rule as it takes into account the number of samples and the spread of the samples. However, there are some shortcomings of using histograms to show distributions (namely the bin count and width, and the distribution's maximum and minimum). Therefore,

in an attempt to combat these problems, the cumulative distribution plot for the data is also displayed on the graphs. The data used in these plots is from only the design candidates where the maximum number of pilots per operating base was set to two (and excluded the candidate that had no UAS assigned as this unnecessarily skewed the distributions). However, when the maximum number of pilots per operating base was limited to one, the histograms show a similar distribution of data.

Figure 5.13 shows a unimodal, symmetric distribution (with a few outliers) of the 95% confidence interval width as a percentage with respect to the asymptotic sample means. The percentage error is small with 98.9% of the design candidates having a 95% confidence interval width less than 0.4% of their sample mean at 250 replications. Also, when plotted as the 95% confidence interval width (*i.e.* not as a percentage of the sample mean) the distribution formed a normal distribution. Therefore, the outliers are mostly due the design candidates having a lower mission-success score and thus raising the percentage error. From this analysis, 250 replications is deemed an acceptable number.

Figure 5.14, which shows the 95% confidence interval width as a percentage of the sample mean for the total service cost, appears to follow closer to a Poisson distribution[6]. The percentage error is still small with 95.7% of the design candidates having a 95% confidence interval width less than 1% of their sample mean at 250 replications. However, the reason for the distribution shape and the increased number of design candidates with entries above 2% error was investigated.

To investigate the spread of 95% confidence interval width for the total service cost the furthest outlier in Figure 5.14 was selected and studied. This outlier was the design candidate for zero UAS 1 and one UAS 2 based at Bembridge. The title given to this design candidate is $0, 0, 0, 0, 0, 1$ as per the naming scheme described on page 85. One reason for this design candidate being an outlier is due to the fact that the total service cost is the lowest of all design candidates (with only one operating base and only one of the low-cost UAS) therefore any variation in cost due to the stochastic nature of the model will be accentuated.

The convergence study for design candidate $0, 0, 0, 0, 0, 1$ is shown in Figure 5.15 and displays the cumulative mean and the 95% confidence interval of the total service cost against the number of repetitions. Also plotted are the raw costs of each repetition of the simulation for this design candidate.

The raw points plotted in Figure 5.15 show grouping into bands of total cost where the bands increase roughly by the cost of a UAS 2. This indicates that the UAS was replaced due to a loss of the platform a certain number of times where that number is represented by the height (or level) of the band. This is confirmed by plotting the count of the number of times the UAS 2 unit was replaced in each replication as shown in

[6]This distribution shape was also seen when the data entries were not presented as a percentage of the sample mean, but just as the 95% confidence interval width.

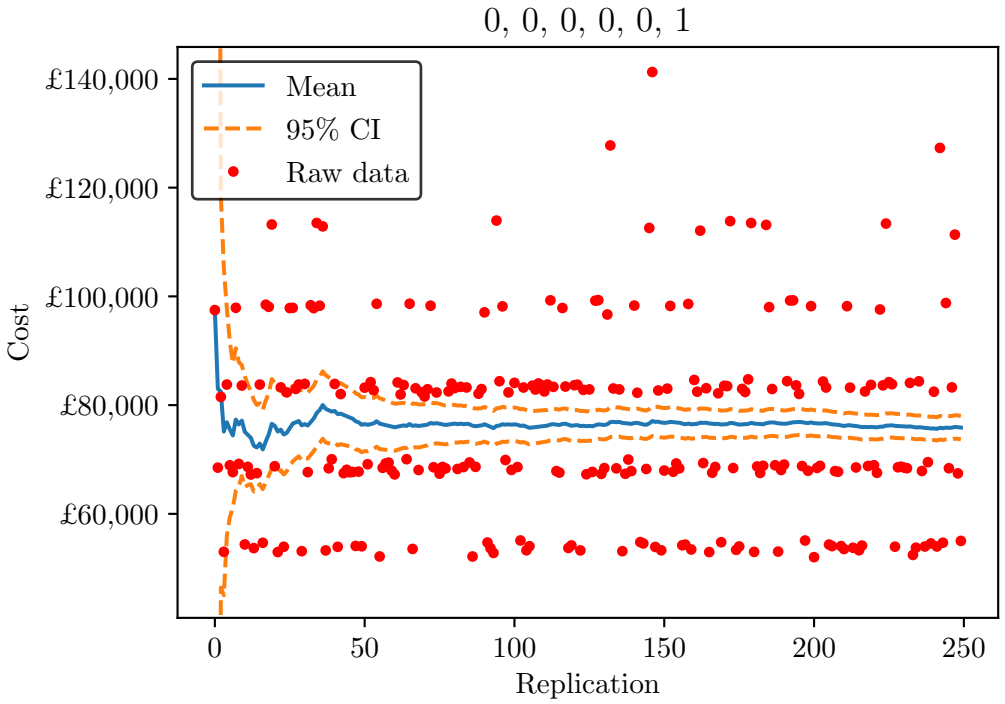


FIGURE 5.15: Convergence study for the total service cost outlier. The cumulative mean and 95% confidence interval is plotted along with the raw data points. Bands of raw data can be seen forming separated on average by the cost of the UAS 2 unit.

Figure 5.16. The total of each bar matches the count of data points in each band formed in Figure 5.15.

There are several possible reasons built into the model for why the loss of a platform could occur. In this study, the loss of platform was due to either a catastrophic component failure (*i.e.* the platform could not continue flying) as determined by the reliability and maintenance model for the UAS, or a discrete adverse change in wind conditions resulting in a depletion of energy (*i.e.* low fuel or battery). In the case of design candidate $0, 0, 0, 0, 0, 1$ which had only one UAS 2 platform (which has reduced range and endurance compared with a UAS 1), there is a greater chance of it making multiple trips to a task location due to needing to refuel, thus increasing the platform's total flight time. This in turn leads to a higher risk of a critical component failure and the higher time in the air also increases the likelihood of the platform being caught out by a change in the weather conditions.

Therefore, the reason for the slightly increased spread in the 95% confidence interval width for the total cost of the service compared with that for the mission success score is because if the design candidate does not perform well at the service (*i.e.* it results in multiple replacements of the UAS units) it can incur large costs. However, these outliers should only be a concern if they are found to be an optimal solution and then more replications might be required to ensure the sample means have tended towards the real

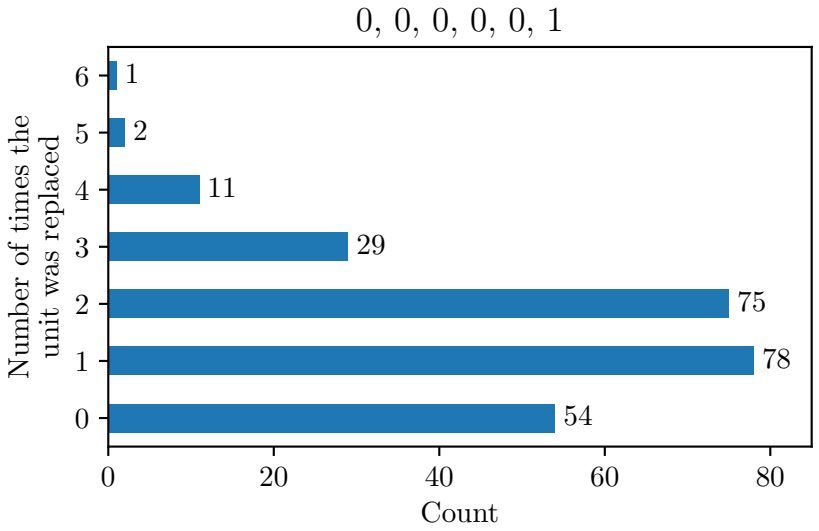


FIGURE 5.16: Bar chart showing the count of how many times the UAS 2 unit was replaced in each replication for the $0, 0, 0, 0, 0, 1$ design candidate for the 250 replications. These counts match the number of raw data points per band in Figure 5.15.

mean. Otherwise, the distribution displayed in Figure 5.14 suggests that 250 replications is also an acceptable amount of replications.

5.2.2 Effect of pilot limiting

One of the investigations in this case study was the effect of limiting the number of pilots available per base. This number of pilots per base was limited to either 1 or 2. The number of pilots was set to follow the total number of UAS platforms at the operating base up to the set limit. If there were no UAS assigned to an operating base then there would be no pilots assigned to it either.

The mission success score is plotted against the total service cost in Figure 5.17 with the data points categorised by the pilot limit. This is also the first time the full set of design candidates have been shown on the bi-dimensional graph that will be used to find the optimal design via the value function. From Figure 5.17 we can see the limit of feasibility form, where to increase the mission success score, the total service cost must increase. This is where the optimal design will sit because the value function used in this case study only uses these two parameters. However, if another parameter was introduced to the value function, for example the stakeholders want to reduce the amount of fuel used (*i.e.* favour electric over petrol), then, when plotted as a three dimensional graph, the optimal design will sit on the limit of feasibility surface. Beyond three dimensions it is hard to effectively communicated this visually to the end-user. One suggestion is through pair-wise plots. The data set plotted in Figure 5.17 will be used in the background of the plots used for further analysis to highlight the overall data set.

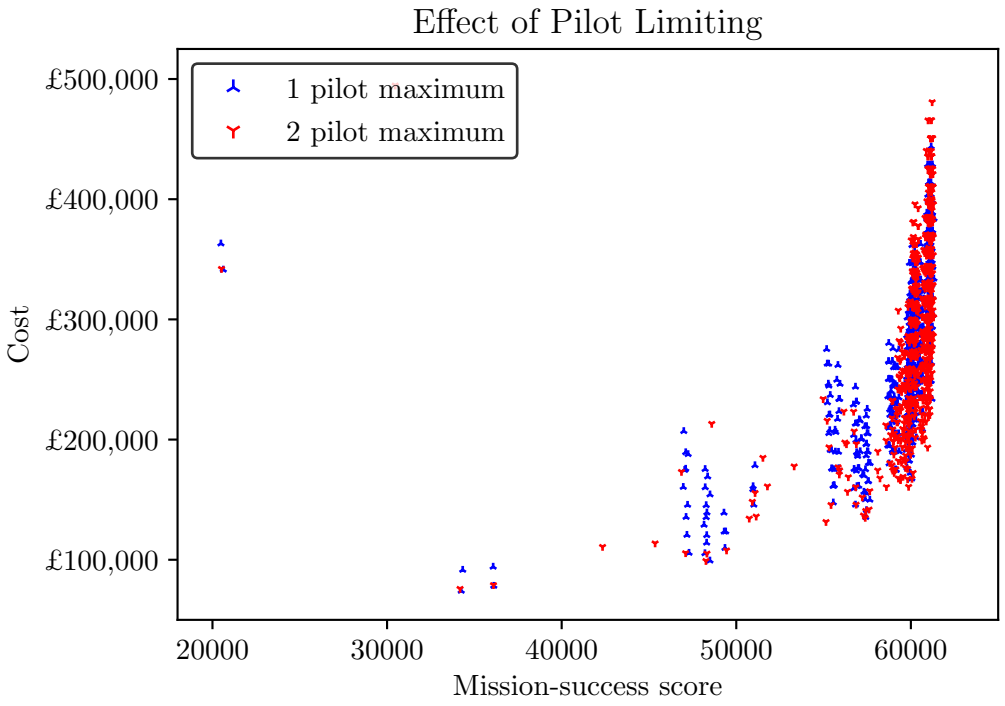


FIGURE 5.17: Plot of mission-success score against total service cost for all design candidates. The design candidates are categorised in the plot by the maximum number of pilots available at each operating base.

Figure 5.17 shows that by limiting the number of pilots to only one per operating base has a significant effect on the mission-success score. This is demonstrated by the spread of blue points along the x-axis compared to the grouping of the red points at the upper-end of the x-axis. The plot also shows that there is an upper limit to the mission-success score. This suggests that the design candidates that sit on the right-hand-side of the graph cannot improve the mission-success score any further. This could be due to the number of pilots being limited to two, or it could be because the design candidates have achieved the maximum mission-success score available for the service.

Another interesting feature that emerges from Figure 5.17 is the *towers* that form (*i.e.* similar mission-success score but increasing cost). These are most notable in the design candidates that are limited to one pilot. Further analysis of the data set led to the plots shown in Figure 5.18.

Figure 5.18 pulls out the design candidates that only use one operating base and plots them on separate graphs graphs for the different bases (*i.e.* Lymington, Swanage, Bembridge). Within each graph the design candidates are categorised by the pilot limitation (using the data point symbol) and the number of each UAS assigned (using the colour of the data point). The legend can be read where the first number represents the number of UAS 1 and the second number represents the number of UAS 2.

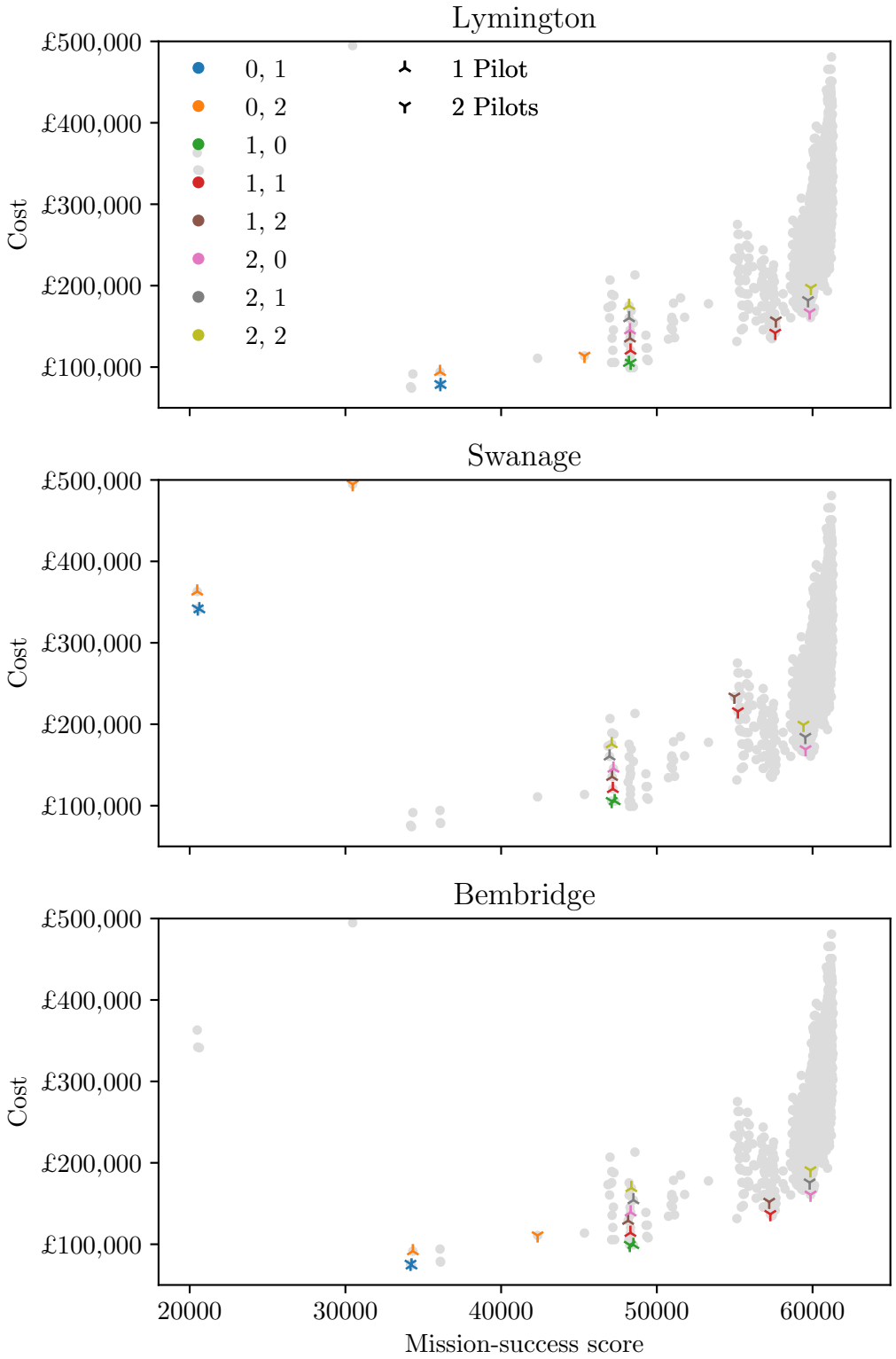


FIGURE 5.18: Plots of mission-success score against total service cost for the design candidates that use only one operating base. Design candidates are categorised by the pilot limitation (using data point symbol) and by UAS assignment (using data point colour) where the legend entries can be read as number of UAS 1, number of UAS 2. The grey markers display all the design candidates.

The first observation to note from Figure 5.18 is that at all three operating bases, the design candidates that only have UAS 2 units assigned to them (the blue and orange markers) perform poorly in terms of mission-success compared to all other design candidates plotted. The reason behind this is discussed earlier in Section 5.2.1 and points to the range and endurance of this platform not being able to service the entire region effectively. This is most notable in the Swanage plot and the high cost indicates high flight time caused by the need to refuel and multiple units replaced. Swanage is the furthest operating base from the daily task and this is potentially a major contributor to this.

All three plots in Figure 5.18 show that the cost and mission-success score for the single UAS options (indicated by the blue and green data points) are unaffected by the pilot limitation (*i.e.* both the 1 pilot and the 2 pilot data points are on top of each other). This was expected as, if there is only one UAS at the operating base, there will only be one pilot.

The investigation into the formation of the *towers* found a flaw in the resource allocation formula. The *towers* form above the data point for one UAS 1 unit (the green data point) at each operating base and increase in price, but each data point in the *tower* shows no significant change in mission-success score. This occurs for all design candidates that have at least one UAS 1 unit and are limited to one pilot. From these plots, this suggests that the missions are always allocated to the same UAS unit regardless of the range or endurance expected for the mission and therefore there is fault in the resource allocation. This theory was confirmed by the analysis of the resource allocation data set that stated that only the first UAS unit listed at the operating base was allocated. This is speculated to also be the case when the number of pilots limit is raised to two. However, in this case, the second UAS is able to be operated by the second pilot while the first is still active. Therefore, these design candidates are capable of responding to simultaneous missions. This resource allocation formula flaw is discussed further in Section 5.4 and a potential solution to the problem is presented.

The final observation from Figure 5.18 is that the introduction of a second platform with the availability of two pilots significantly increased the mission-success score. This improvement is increased by the capability of the second UAS[7]. This is demonstrated by the distinct grouping of the design candidates based on their combination of UAS units and where, as the number of UAS 1 units increase, so does the mission-success score increases.

Overall, these results show that limiting the number of pilots does have a direct effect on the mission-success score. This is because having the availability of a second platform

[7]The author believes this observation is not affected by the flaw found in the resource allocation formula. However, this cannot be confirmed until the simulation is repeated with a new formula.

with a pilot unlocks the ability to attend simultaneous missions and thus is of great benefit.

5.2.3 Dual operating bases

The next set of design candidates that were analysed were those that used two operating bases. This analysis was limited to candidates that had up to 1 of each type of UAS type per operating base. This limitation produced 9 candidates for each pair of operating bases. Therefore a total of 27 design candidates are considered here. These are shown in Figure 5.19 where the pair of operating bases is distinguished using the data point symbol. The permutation of the UAS assignment is signified using the data point colour.

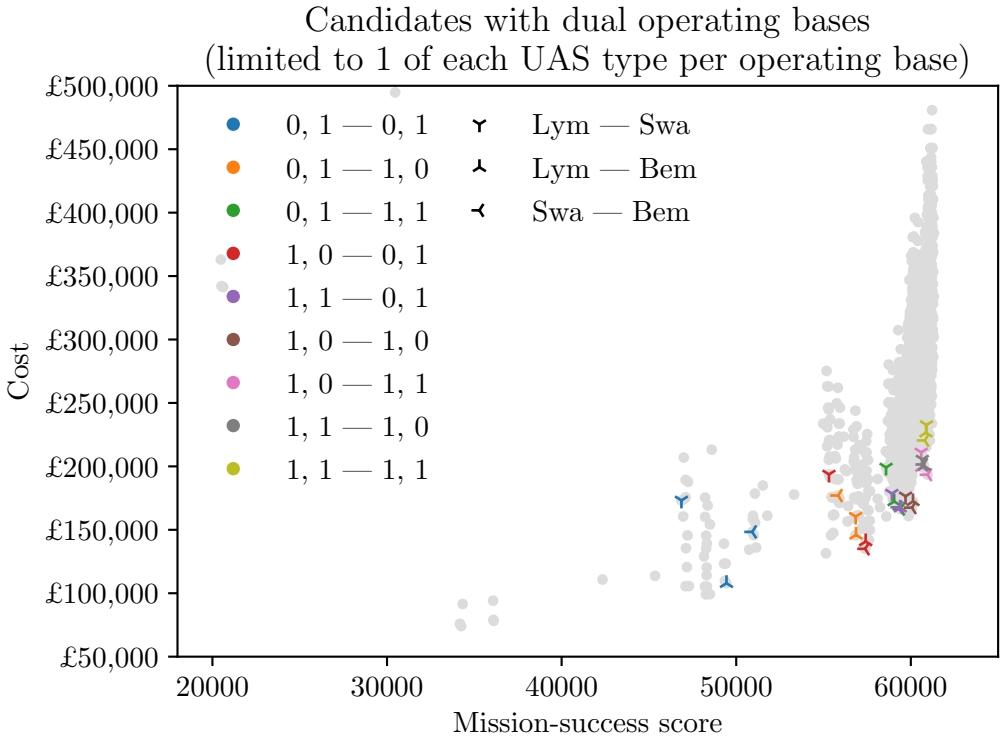


FIGURE 5.19: Plot of mission-success score against total service cost for the design candidates that use two operating bases and are limited to one of each UAS type per operating base. Design candidates are categorised by the pair of operating bases (using data point symbol) and the UAS assignment (using data point colour). The UAS assignment is read as UAS 1, UAS 2 at the left-hand operating base followed by the — and then UAS 1, UAS 2 at the right-hand operating base. The grey markers display all the design candidates.

The plot continues to highlight that only using the less capable platform, UAS 2 (blue data points), the mission-success score is not as high at those candidates that include a UAS 1 unit. However, by spreading the two UAS 2 units across the region (*i.e.* not just using one operating base as shown in Figure 5.18) the mission-success score is improved.

Figure 5.19 also shows grouping where the total number of each UAS in the service is the same. For example, the orange and red data points both have one UAS 1 and one UAS 2. However, the permutation of which UAS is assigned to which operating base has a significant effect. This is highlighted the most by comparing the orange and red data points of the Swanage and Bembridge design candidates. Here, it is shown that it is more beneficial in both parameters (cost and mission-success score) to place the UAS 1 unit at Swanage and the UAS 2 unit at Bembridge.

The combination of operating bases that produced the more favourable design candidates was Swanage and Bembridge (where favourable is considered here as a lower cost against mission score). This appears to be followed by the combination of Lymington and Bembridge. Therefore, this shows it is likely that an optimal design candidate will include Bembridge as one of the operating bases (especially when considering dual operating bases). Logically, these findings make sense as the daily task is performed near Bembridge and ensuring maximum coverage of the region makes choosing Swanage a sensible pairing.

The flaw in the resource allocation is not so significant here as the resources are spread between operating bases therefore the 'closest unit' policy will be in effect. Also, the operating bases are not limited to one pilot so simultaneous missions will be available from the nearest operating base if there are two UAS units allocated.

5.2.4 Optimal designs

The value models described in Section 5.1.4 were used on the design candidates in order to find the optimal design with respect to the stakeholders' requirements. The first value model put more emphasis on maximising the mission-success score by increasing the weighting of the mission-success score in the CBA. The result of this is shown in Figure 5.20.

In Figure 5.20 the value gradient, shown by the colour map, indicates the direction of improving value, where blue represents a low design value and red represents a high design value. It ultimately reaches a maximum on the design candidate that places two UAS 1 units at the Bembridge operating base with two pilots available. This service design uses two of the more expensive and more capable UAS units. However, it reduces costs by locating them at one operating base. The selected operating base, Bembridge, was found through the earlier analysis (refer to Figure 5.18 and Figure 5.19) to be the more favourable operating base.

Figure 5.21 shows the result of the second value model where the emphasis was on reducing cost. This was achieved by reducing the the weighting of the mission-success score in the benefit component of the CBA and thus giving the cost more influence on the design's value.

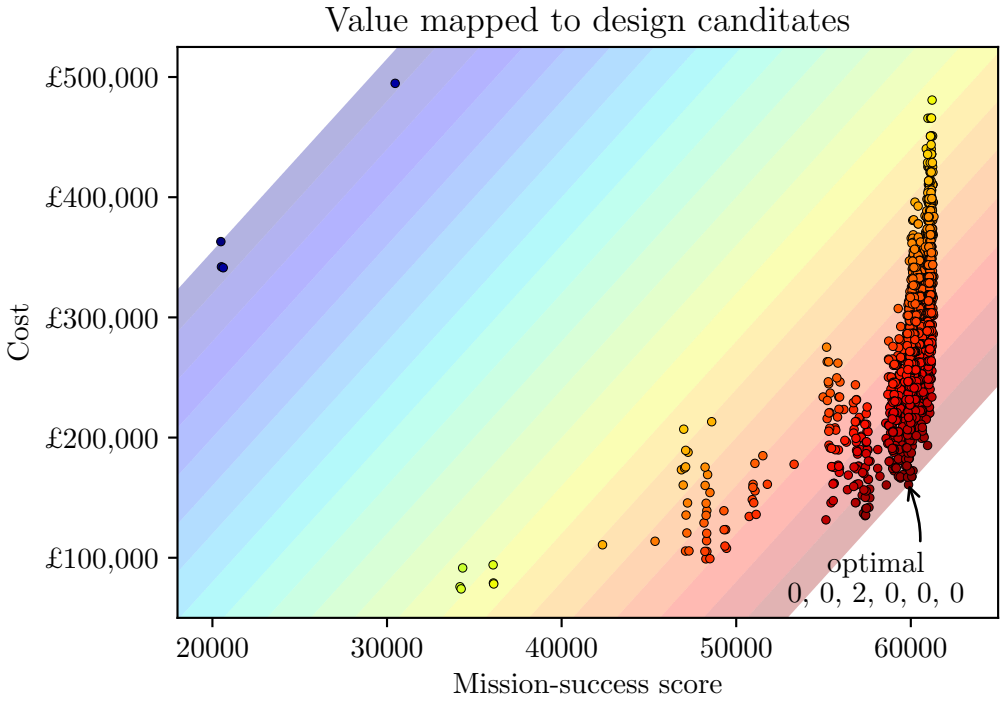


FIGURE 5.20: Plot of mission-success score against total service cost for all the design candidates highlighting their design value by the colour scale where blue represents a low design value and red represents a high design value. The value model is set to promote mission-success score. The value gradient is included to indicate the direction of the value model. The optimal design is annotated on the graph and was found to be the design candidate with two UAS 1 units based at Bembridge with two pilots available.

The value gradient is displayed in Figure 5.21 in the same way as in Figure 5.20. By comparison of the two figures, it can be seen that the value gradient is shallower in the graph for minimising cost (Figure 5.21). This leads to the design candidate with the maximum value being a different candidate. However, it still is found on the Pareto front (also previously described as the limit of feasibility). The optimal design candidate for a value model minimising cost was where one UAS 1 unit was operated from Swanage and one UAS 2 unit operated from Bembridge.

Both of these optimal design candidates make logical sense. However, without the data obtained by this model and the simulation, it would be extremely difficult to justify which matched up to the stakeholders' needs or how they compared. The other advantage demonstrated here is the ability to find an alternative design if the stakeholders' needs changed. Simply by adjusting the value function to suit the new needs of the stakeholders, a new optimal design can be found.

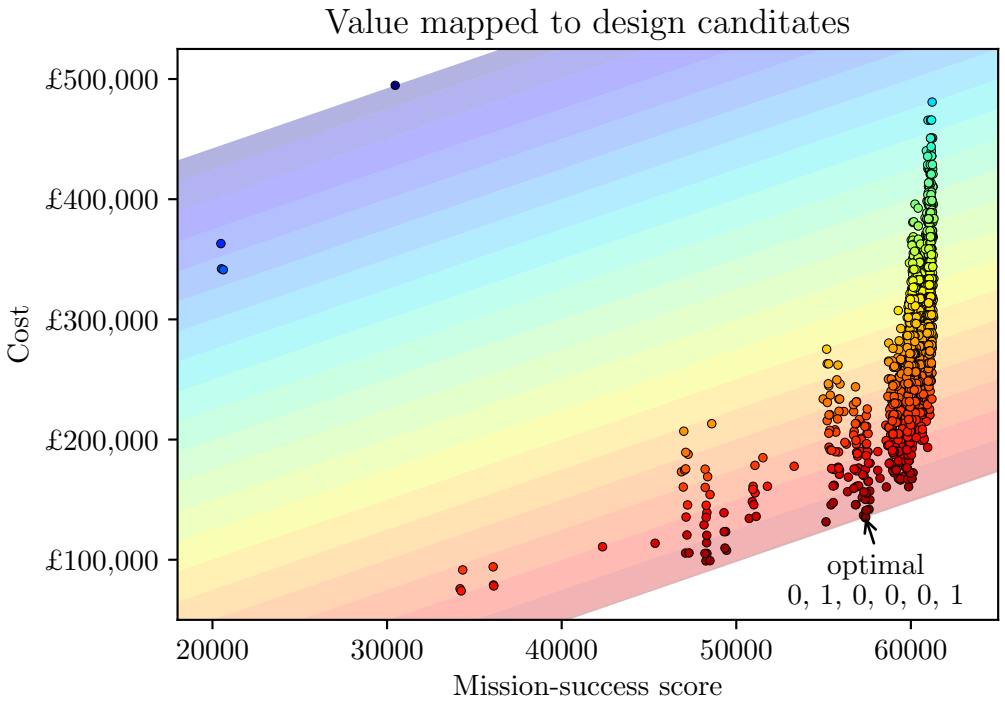


FIGURE 5.21: Plot of mission-success score against total service cost for all the design candidates highlighting their design value by the colour scale where blue represents a low design value and red represents a high design value. The value model is set to promote minimising the total service cost. The value gradient is included to indicate the direction of the value model. The optimal design is annotated on the graph and was found to be the design candidate with one UAS 1 unit based at Swanage and one UAS 2 unit based at Bembridge.

5.3 Benchmarking the model

The model required benchmarking using an independent algorithm as a means to systematically compare the solutions produced and provide confidence in the model. Due to the model developed in this thesis being an amalgamation of UAS performance models, discrete event models, value centric design implementation, and asset allocation algorithms, (and therefore creating a novel model) it is difficult and challenging to benchmark the entire model against an independent algorithm without heavily tailoring the algorithm to the problem and thus losing its independence. A review of academic methods that may be employed to best benchmark a new model found that the research performed on computational tools in biological and biomedical sciences are often subject to benchmarking and this has generated a large number of publications discussing the methods, issues, pressures and standards towards benchmarking [138, 139, 140, 141, 142]. In these academic outputs advice is given as to how to handle the introduction of a new method or model: it is suggested the focus of the benchmark should be on evaluating the relative merits of the new method. However, some advantages, benefits or sophistication of the new model might fall out of the scope of the benchmark. For this reason the benchmark needs to be carefully designed to ensure it is a fair comparison [138].

In the case the model presented in this thesis, it was decided to focus the benchmarking on the resource-allocation and variable-costing elements of the model, as these were the key components that drove the solution. The independently developed algorithm selected as a benchmark model was from the open source OR-Tools[8] optimisation software suite which is developed by Google and is tuned for tackling vehicle routing problems, network flow problems, integer and linear programming, and constraint programming [143].

The selected benchmark algorithm was based on the Capacitated Vehicle Routing Problem (CVRP) where the capacitated value (also known as demand in the CVRP model) is the distance travelled during the task of the mission. Due to the type of UAS service the presented model in this thesis was designed handle, the CVRP was set up as a many depot (operating base), single customer (mission) problem. The vehicles were modelled to have the capacity matching that of the range of the UAS. This allowed the CVRP algorithm to state if the vehicle could complete the mission. This was similar to how the presented model operated. The CVRP solver used the *path_cheapest_arc* as its first solution strategy. It also applied the *guided_local_search* strategy to allow the solver to escape potential local minimums to find a better solution. The mission demands and the distance matrix, describing the distances between nodes (each operating base and the individual missions), used in the CVRP model were calculated from the output of the presented model's mission generators (taking into account a static version of the UAS and payload performance models).

The inputs and methods described above created a simplified resource allocation solver that could be used to compare the findings with the presented model. The main simplifications of the implemented CVRP model were that: (1) it was not time-based and therefore it was not set up to dynamically update the calculated distances due to changing weather conditions, and it was not aware of missions overlapping in time; (2) if none of the UAS platforms had the range to complete the entire mission, it would not allocate an asset to that mission, whereas the presented model would provide a UAS platform to complete some of the mission task within reason or perform multiple trips to complete the entire mission task (a policy set by user input); (3) the CVRP model had no component-reliability model which meant that all platforms modelled were available throughout the service - this is unlike the presented model where maintenance tasks or platform loss would temporarily remove a UAS from availability (again, another time-based element of the presented model). It was possible to run the solver to find the best UAS for each mission and compare the resource allocation to the presented model, but the simplifications needed to be taken into account when interpreting the results.

An initial comparison took place for replications of the design candidate $1, 1, 1, 0, 0, 0$ (*i.e.* a single UAS 1 at each operating base). Of the total number of missions within each replication in the data sets, 90.90% resource allocations were the match between

[8] See `https://developers.google.com/optimization` for more information about OR-Tools and how to use it.

the CVRP and presented model, leaving 9.10% different. These were analysed to see the reasons for deviations. Firstly, it was found that 2.49% of the missions were classed as not possible by the CVRP model due to it not having the functionally to allow partial completion or multiple trips. Of the remaining different results, 6.57% were found to be due to missions overlapping in time in the presented model and therefore the list of available resources was reduced and did not include the otherwise optimal UAS (*i.e.* as it was already performing another mission). This left the final 0.05% difference between the two models unaccounted for. On investigation it was found that the mission location fell almost equidistance between two operating bases in these cases and therefore the difference in resource allocation was most likely to be due to the weather conditions which are taken account of in the presented model but not in the CVRP model. This refinement in the presented model would account for why the geographically more distant operating base would be favoured. These results are visualised in Figure 5.22. The majority of the results provided a good match but where differences occurred they were due to the limitations of the implementation of the benchmark model and could reasonably be expected. This is of no criticism to the CVRP model.

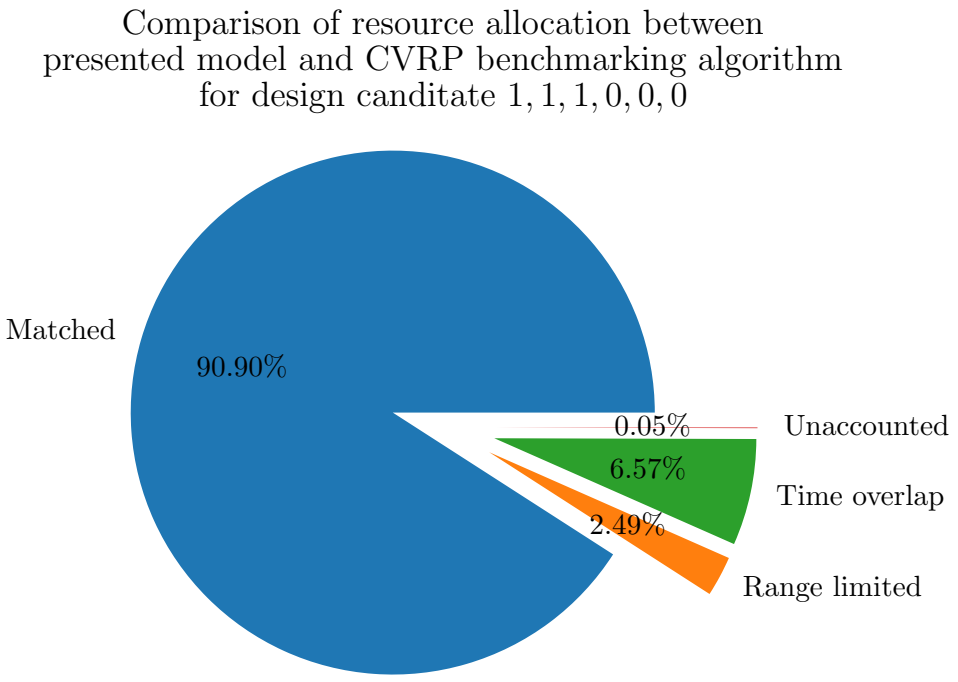


FIGURE 5.22: Pie chart showing the comparison between the resource allocation results of the benchmark CVRP model and the presented model for the $1, 1, 1, 0, 0, 0$ design candidate (a single UAS 1 at each operating base). The values shown are calculated as a percentage of the total number of missions per replication and then averaged over the data set.

As further design candidates were replicated, it was noted the resource-allocation matches between presented and benchmark models reduced as the number of UASs decreased, and/or as the number of UAS 2 platforms increased. This was due to the reduced range

of the UAS 2 platform resulting in either more *range limited* differences or the CVRP model selecting a UAS 1 due to its improved range capabilities. This effect of this can be seen in the comparison of models for the design candidate $0, 1, 1, 1, 0, 0$ (*i.e.* one UAS 2 platform at operating base 1, and one UAS 1 platform at each of operating base 2 and 3). The results are shown in Figure 5.23. In this example, deviations between the two models can be seen regarding resource-allocation due to the range limitations of UAS 2 and the simplifications made in the benchmark model. The CVRP model provided different resource allocation solutions for 9.07% of the missions due to lesser capabilities of the UAS 2 and the model not allowing partial completion or multiple trips.

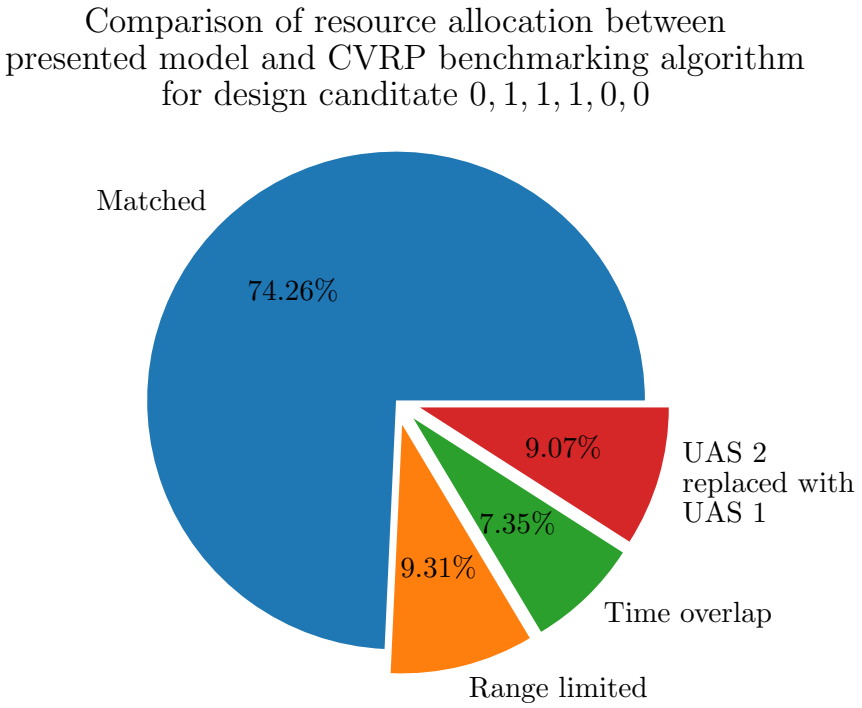


FIGURE 5.23: Pie chart showing the comparison between the resource allocation results of the benchmark CVRP model and the presented model for the $0, 1, 1, 1, 0, 0$ design candidate (one UAS 2 at operating base 1 and a UAS 1 and the others). The values shown are calculated as a percentage of the total number of missions per replication and then averaged over the data set.

The chosen benchmark model was not wholly suitable due to the limitations described, but with further research it may be possible to find a more adapted variant of the CVRP that better represents the real-world application of a UAS service. This would be expected to result in a better comparison to the presented model. The adaptations required would need to cover the ability to fragment the mission task, handle time-overlapping missions, and account for the stochastic variables (*i.e.* component-reliability and weather conditions). However, the benchmarking exercise highlighted some positive points relating to the presented model. Firstly, the comparison analysis of the two models was relatively easy to conduct due to the transparency of the results produced by the presented model. With the wide spectrum of data produced and recorded it was possible

to drill down into the data to categorise the deviations and confirm the hypothesised reasons for them. Secondly, the benchmarking exercise demonstrated the effects and benefits of the complexities built into the presented model which enabled it to better represent the real-world application. The presented model was developed in a modular, generic, and graphically-enhanced way and produced rigorous results in an uncomplicated manner.

5.4 Summary

This chapter has demonstrated the capabilities of the model as a UAS service design decision support tool. It has shown that using data sets that describe the service tasks and regional weather it is possible to create model inputs that define the mission and weather generators. It was also shown that the input of UAS parameters and the location of the proposed operating bases allows the designer to create a large pool of design candidates. Finally, after running the simulation for the number of replications that provides a sufficient estimate of the output mean, it is possible to analyse the performance of the design candidates. Through the use of a value function that represents the stakeholders' needs it is possible to obtain an optimal design for the service. The full collection of output data produced by the simulation allows the designer to understand the reasoning behind the choice. It can also allow the designer to see areas of high cost or risk.

Through the analysis of the results produced in this case study, a error in the resource allocation function was identified. This presented itself when there were two or more UAS units stationed at one operating base and was highlighted when the number of pilots was limited to one. The resource allocation function defaulted to always use the first UAS unit listed at that operating base that was capable of attending the mission. This did not affect design candidates where multiple bases had an individual UAS unit as the resource allocation function had a policy set to choose the closest UAS unit to attend. In this case study, if the operating base had a UAS 1 assigned, then this would be the first UAS unit in the list and therefore would be the first choice (*i.e.* because of UAS 1's greater range and endurance compared to UAS 2, if it was not capable of attending then the UAS 2 unit would also not be able to). Therefore, to improve this the intra-operating base UAS allocation should be a settable Concept of Operations (CONOPS) policy. This could then be used to select the UAS unit that is not excessive for the task. For example, this policy could sort the capable units in order of range or cost per flight hour.

This case study also highlighted the importance of setting appropriate CONOPS and other policies for each UAS unit. During the convergence study in Section 5.2.1 it was found that the increased 95% confidence interval width for a design candidate with a

UAS 2 unit was mostly brought about due to losing the UAS unit. These losses were due to both critical failures and weather related incidents.

An example of setting an appropriate CONOPS policy in this case study can be demonstrated by considering the reserve fuel. The overall UAS service had a global policy in place regarding reserve fuel. This policy was used to define which UAS units were capable of attending and performing the task (by calculating the range without including the reserve fuel amount in the range equation) and when the UAS unit should return to refuel if the mission was not yet complete. If the reserve fuel amount (stated as a percentage of the total fuel) was set high, then the UAS would have a reduced operational range and hence could potentially require more refuelling trips to complete the mission, thus increase the consumable and operational costs. This increase in flight hours also increases the chance of a component failure (which leads to increased maintenance cost or critical failure). However, this fuel reserve might prevent the losses caused by the discrete change in wind that occurs in the simulation (*i.e.* the UAS will have reserved range to overcome an increase in the effective range caused by an unfavourable wind direction and speed).

Alternatively, if the reserve fuel amount was set low then the number of refuelling trips would reduce and hence the number of flight hours would reduce (*i.e.* fewer transits between base and mission location). This, in turn, would result in fewer critical failures as they are based on the mean time to failure metric which references the number of flight hours. It would also have an effect on the number of maintenance operations and cost as this is also reliant on the number of flight hours. An improvement to the simulation would be to move the fuel reserve amount from a global policy and introduce it at a UAS unit level policy.

The effect of adjusting the CONOPS was not part of this case study, but is possible through the model and simulation presented in this thesis. However, by increasing the number of variables to the simulation, it also increases the number of design candidate produced. The computational time to run all 1458 design candidates presented in the case study took in the region of 12 hours using a high performance desktop[9]. The results can be visualised quickly. However, a further 24 to 32 hours of work is required for full analysis and final quantification of the design decisions. Therefore if a CONOPS study is required, it is suggested that the service design optimisation is completed in stages to reduce the number of design candidates. This can be achieved by completing an exhaustive search to find the design candidates that sit on or near to the Pareto front (*i.e.* optimal designs and design alternatives). From here, a CONOPS study can be performed on this limited pool of high performing design candidates.

[9]The desktop used to run the simulations had 4 cores, 8 logical processors, 3.60 GHz CPU and 32.0 GB of RAM. The simulation's replications utilised parallel computing through the AnyLogic software.

Chapter 6

Discussion and conclusions

This chapter starts by providing a summary of the work presented in this thesis in Section 6.1. This is followed by Section 6.2 which addresses the broader issues around the tool development and the key findings from this research. The model's complexity, transparency and computational measures are discussed. Alongside this, the scalability with operational size and scope, asset heterogeneity and applicability to a variety of business-models is commented on. The research aims and objectives are revisited in Section 6.3 and discussed. Finally, Section 6.4 presents suggestions towards further and future work to develop and refine the model, extend the research and build on the achievements to date.

6.1 Thesis summary

This thesis has described how a UAS service can be modelled such that design candidates for the requested service can be compared and an optimal solution found. This has been achieved by the use of discrete event based simulations of the missions and tasks of a UAS service and modelling how the UASs respond. The model captures accurately the performance and reliability of the UAS platforms and the performance of EO payload sensors. It also takes into account some real-world factors that influence the UAS's response to the tasks, for example the geospatial nature of the UAS service and the local and temporal wind conditions. The model was designed in a modular way that allows it be further developed and new features added. It also included an animated view of the simulation which can be used to provide confidence in the model setup and debug any problems arising from the logic of the model. This can be turned off to save the computational cost when completing batch runs.

A case study has demonstrated how the service details which are used to describe the missions and tasks in the simulation are formulated. A large set of design candidates

(generated by varying the number, type and base location of the UASs) was evaluated through the simulation and their results were analysed. The presentation of the outputs gave a clear rational for the results allowing them to be traceable and understood. The results of the case study produced sensible and logical solutions, and importantly, they provided auditable evidence of their costs and effectiveness.

However, as with all models and simulations, the model created is only as good as the inputs provided. Therefore, the fidelity of the model was set to match the expected level of detail in the inputs. This benefited the computational cost of the model by not including unnecessary levels of detail.

It became apparent during the design of the model that a comprehensive list of CONOPS and policies needs to be drawn up and agreed as any omissions could lead to a less effective optimal solution. For example, in the case study the policies that could have lead to a different optimal solution were: the amount of fuel reserve to account for emergencies (*e.g.* an adverse change in wind conditions); and the selection of the platform type best suited to the mission (*i.e* not selecting an overly capable platform which would increase costs and remove it from the pool of platforms available for subsequent missions requiring its capability).

6.2 Discussion

6.2.1 Tool development

In Section 3.1, the four high level requirements for the framework of the decision-support tool were stated. These were that the model produced should be comprehensible and simple, generic, modular, and realistic. The adherence to these principles throughout the design and development of the model greatly helped bring together the multitude of disciplines and elements of the complex real-world application into a relatively succinct model. The requirements also provide key points to discuss the general lessons learnt during the tool development.

The modular design of the model meant that additional extensions could be turned on or off (*e.g.* the inclusion or exclusion of the weather conditions). This served as a way to test the base model and the effect of each individual extension in isolation and as a whole. It also allowed the overall realism of the model to be improved and adjusted. From the experience gained by designing a model of this scope, it is highly recommended to follow a modular approach as it breaks the big picture into manageable portions, each of which can build up the complexity and realism as required but still have a functioning base model. The challenges of implementing modularity in this thesis stemmed partially from defining the boundaries and level of detail for each module and then integrating them into the base model. The intuitive approach to the boundaries was to align them

with the physical boundaries being modelled. This was progressed by forming a tree-like structure of modules and sub-modules to reduce the number of integrations required with the base model – the act of integrating is often a source of error and programming bugs, and therefore minimising the number of times this has to be performed is favourable. For example, the weather module was categorised as its own entity which then had the capability to contain multiple sub-modules (*e.g.* the wind conditions). This aided the integration process as only one module was required to be integrated into the base model, rather than several modules defining the individual weather elements.

Another challenge in the development of the model was the need to make the model comprehensible and simple. This was facilitated by keeping the inputs and outputs at the forefront of the design. Thus, the level of detail and knowledge the user had to supply for the model to produce meaningful and acceptable results was constantly assessed. The same approach was taken for the results being produced by the model with focus on ensuring these were transparent (*i.e.* the raw values produced were available alongside the calculated outputs so that the results were traceable and the logic was clear). By creating traceability within the outputs of the model, the process of analysing and evaluating the results was made less arduous and more manageable. This means that the key factors influencing the results were easier to identify, and further studies could be performed where required (*e.g.* a sensitivity analysis of a CONOPS policy). It was also imperative throughout the development of the model to consider and constantly re-evaluate the assumptions and limitations of the model (and the additional modules) and provide this to the users to ensure the limitations present are understood.

Developing a model capable of accepting the vast spread of UAS types and UAS service types as the inputs and descriptors was one of the most challenging elements to the tool design. As the model increased in sophistication, the interactions between modules became more complex. In some aspects, the scope of the tool presented in this thesis was restricted in order to ensure the remainder of the tool could be developed and the tool as a whole could be demonstrated (*e.g.* the focus of service types was restricted to UAS only services and did not include a ground-air asset mix in which a UAS could supplement the assets of an existing service). The consequence of the restrictions is a reduced genericity of the presented model and thus the range of applications to the end users at its current level of development is also affected. However, because the other high level requirements (modular and comprehensible) were followed in the model development, the updates required to include these additional elements would be straightforward to incorporate. It is, though, important to ensure that any increase in the scope and capabilities of the tool does not reduce the quality of the results nor the confidence in the tool or its ease of use.

Finally, the use of an animated simulation during the development, debugging, discussion, and presentation of the model was invaluable. This vastly reduced the time to resolve bugs and programming errors as the entity flow in the discrete event simulation, or asset location in the geographical view could be used quickly to confirm or refute that the

model was working correctly. This highly beneficial view of the simulation and model was made possible through the use of an off-the-shelf programming environment, AnyLogic. It was also crucial to be able to turn off the visualisations as they are computationally expensive and unnecessary once confidence is gained in the model and its setup and batch runs are required. Although it does require additional time and effort to implement the animations and graphical representation, the advantages pay dividends.

6.2.2 Application of tool to different UAS service types

The decision support tool was designed to model new UAS services that operate from fixed locations and serve missions individually using a mixed fleet of UASs. This style of UAS service is typically a surveillance/searching based service or a rapid/urgent delivery based service. However, the availability and capability of UASs has advanced over the past decade and new styles of services are emerging. The new types of services include: (1) logistical delivery services solely using UAVs; (2) logistical delivery services with UAVs integrated with other ground-based delivery vehicles - both (1) and (2) would often requiring route planning in the mission-task phase; (3) mobile operations that do not necessarily require a fixed base (*e.g.* travelling to different locations to undertake aerial surveys).

The implementation of mobile operations would require minimal changes to the current framework. This would likely be through an additional module describing the new mission logic.

The logistical delivery services solely using UAVs could readily be modelled through the framework setup in this thesis. The AnyLogic programming environment used for the model has some built in libraries for routing and is capable of adding external/independent VRP solvers to the Java library attached. This and the multi-modal delivery service (*i.e.* a service being delivery with both UAV and ground-based assets) would be an interesting direction of further research.

In the case of modelling a multi-modal service, the user of the tool is no longer just interested in how the mix of UAVs perform, but how the integration of the UAVs into the full service affects the business as usual (BAU) case. Therefore, to introduce multi-modal services into this framework such that they can be compared to no-, partial- or full-integration, the model will have to be capable of modelling and interacting with the BAU service. This creates the need for more complex user inputs and setup requirements, and in turn moves the model away from ease of use and genericity. However, this does increase the range of applications the model can handle, and including the base-line BAU solution helps compare the effect of integrating UAVs into the service.

It is worth noting that the preliminary analysis stage of a multi-modal service might not require the fidelity provided by the presented model (*e.g.* stochastic weather conditions,

component level reliability, and dynamically calculated range equations). Instead, it could be adequately served by a bespoke lower fidelity Fleet Size and Mix (FSM) VRP solver to find a base-line UAS service level. The results of this could then be studied via the presented model to, firstly improve the accuracy of the costing and value achieved by the UAS, and secondly explore different CONOPS and policy settings.

6.2.3 Scalability and applicability of the presented model

The presented model in its current form has the complexity class $\mathcal{P}$ (in terms of time complexity) due to the resource-allocation currently being a decision-type problem rather than an optimisation problem. The UAS services which the current tool has been designed to model has no internal optimisation problems present, and therefore will scale as a polynomial time-complex problem with the increase of inputs (*e.g.* missions, decisions, and resources). If, however, each mission required optimisation (*e.g.* included a VRP) then the model would be $\mathcal{NP}$-hard. This means it will scale exponentially with the inputs relating to the optimisation problem. See Hoos and Stutzle [144] for an introduction to computational complexity in combinatorial problems.

For the case study presented in Chapter 5 the 250 replications of each design candidate (to account for the stochastic inputs) took on average 33.5 seconds. Therefore, to run all 1458 design candidates on one processor the simulation would take 48,843 seconds, which is equivalent to 13.57 hours. This represents when the iterations (*i.e.* the set of replications for each design candidate) are solved in series. However, this time can be reduced by running each iteration on parallel processors. The computer used during this case study had 8 logical processors (4 cores), 3.6 GHz CPU and 32 GB of RAM. When overheads (other processes affecting the computers performance - here estimated to be an additional 20%) are taken into account, the total time to run the entire model for all design candidates is theoretically around 2 hours. However, in practice, it took in the region of 12 hours due to running the iterations in batches to ensure the data collected was complete and uncorrupted, and to mitigate the risk of a computer error or failure resulting in a loss of data by backing up the data incrementally.

The optimisation study of the UAS service (i.e the choice of operating bases and assign mix of UAS) presented in the case study is a $\mathcal{NP}$ problem. In this thesis, the optimisation study was tackled by performing an exhaustive search over all the design candidates. This was made possible by limiting the number of permutations with repetition to manageable numbers. For example, if an extra operating base was included in the study the total number of design candidates (ignoring the pilot limitation study) would increase from 729 to 6561. If an exhaustive search was to be performed on all design candidates it would in theory be expected to take 9 hours (including overheads). This equates to 54 hours if the same 1:6 ratio of theoretic to in-practice time is applied.

This poor scalability means that the exhaustive search method used in the presented model is may not be a realistic proposition for larger problems. Therefore, to improve the applicability of the optimisation study using the model, it is suggested that the use of optimisation solvers, such as those employed in VRPs or scheduling problems, is explored. These can be applied as an optimisation wrapper to the current model, with the aim of minimising the value function by varying the permutations of the inputs. One such approach is the branch and bound algorithm as it is used for discrete and combinatorial optimisation problems [145].

Overall, the model in its current form is scalable and applicable to many different variations of stationed UAS services providing single task missions. This makes it very suitable for cost and value sensitivity analysis and exploring the service's CONOPS and policies. Should the model be required to perform a large design candidate search than that presented in the case study, then a suitable optimisation solver should be used on the outputs of the model.

6.3 Review of research aims and objectives

The research aim presented in Section 1.3 was to

> *explore the development and application of a mission-based computational simulation and optimisation environment to have transformational impact on decision-making when designing an uncrewed aerial system service.*

The work presented in this thesis has shown how a mission-based computational model and simulation can be used to test UAS service designs and ascertain their cost and performance. The results from the simulation were used to lead the user to an optimal design through the use of value-centric decision methods. Moreover, the simulation outputs a large collection of data that can be analysed to understand the findings of the model and more in depth information of the design's performance. This allows the reasons behind the optimal solution to be transparent and traceable, thus this can provide confidence in the solution.

The value of the tool lies in the certainty of the results, and the confidence the user can place on the service selected with respect to cost and stake-holders values. This is achieved through the automated assessment of the design candidates and rigorous treatment of all the design variables. However, the extent of the impact on the decision-making has not been quantified in the work presented. Therefore, it is recommended that a validation experiment is performed. This could be achieved by running a workshop session for UAS service design experts where the decision-support tool could be used to model a particular service. The optimal solution can be compared to that found by

traditional methods and the experts can comment on the value of the tool to practices in the field. This would allow the model to be peer reviewed.

The objectives that formed the path of this research were achieved. The UAS was modelled so that it could be simulated in terms of its performance, cost and reliability in a discrete event model. The current research landscape regarding decision making methodologies was reviewed and a value-centric design methodology was integrated into the simulation outputs. The computational tool was developed to replicate real-world applications and was tested in a specific case study. The capabilities and limitations of the presented model were assessed and discussed along with the applicability and scalability of the model. Also, general lessons learnt were provided to help with the development of similar computational tools. Finally, these discussions led the suggestions of future work and alternative research methods which are presented in the follow section.

6.4 Future work

One of the first areas of future work is to address the error found in the allocation of UASs to missions as described in Section 5.4. A solution to this was proposed and involved the introduction of an intra-operating base UAS selection policy.

It was found during the case study that a large percentage of the design candidates produced were not close to the limit of feasibility. These incurred significant computational cost despite their lack of suitability in many cases being foreseeable. This is a by-product of performing an exhaustive search on a large number of variables each with several choices. If some of these candidates could be eliminated without full analysis, this would reduce computational cost or allow the saving to be used to vary other influential parameters of interest, which may result in an improved optimal solution.

One solution to reduce the number of design candidates is to run a more defined or restricted search which could eliminate the candidates most unlikely to be optimal. Alternatively, future work can be done to find a suitable optimisation method for minimising the design candidates studied, as discussed in Section 6.2.3. For example, the use of a branch and bound optimisation method could be considered as it is used for discrete and combinatorial optimisation problems. This will require the value model to be fixed at the beginning of the study to decide which nodes to branch. Also, care will be required to create the subsets of design candidates such that they do not overly restrict the solution space. By doing this, it removes the possibility to explore easily alternative value models as the solution space is tailored to a particular value model. This would be an interesting area of future work.

The model should be used to explore the effect of different CONOPS and policies, including the flight regime of the UASs. This should be treated as a sensitivity study to determine the influence they have over the value of the design.

To improve the benchmarking of the model, either a more suitable algorithm needs to be selected for the validation, or the limitations of the CVRP model used need to be addressed. Firstly, the policies that are affecting the resource-allocation result in the presented model could be turned off (or avoided in the case of the time-overlapping missions), allowing for a better comparison. However, this disadvantages the presented model as it is not demonstrating it at its full potential. Alternatively, it could be beneficial to research Dynamic VRP (DVRP) and solvers and design a benchmark algorithm based on this. Ojeda Rios *et al.* [81] provides a good survey of the research on DVRP applications and solutions until 2021. The most relatable type of problem is the *dynamic and stochastic* category. If a DVRP model is produced, it would be interesting to compare the accuracy and speed of the presented model and DVRP benchmark algorithm.

Finally, the modules that are considered an important aspect or an influential features of the service that were deemed out of scope in this thesis should be incorporated. For example, a communication range module could be included to model the available communication methods and either switch between them as appropriate for the missions in the service, or limit the UAS to one method of communication and assess the impact on cost and capability. This should also include upgrading the presented modules where required or suggested. For example, the weather module can be upgraded to include other influencing factors, for instance, precipitation. Also, another upgrade could be to incorporate a smoother transition between wind states in the wind model.

References

[1] Cozzens, T. UAV Market to Hit $37B by 2022 - report on RnR Market Research's forecast. *GPS World*, 27(4):35, 2016.

[2] Markets and Markets. Unmanned Aerial Vehicles (UAV) Market. www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html (Accessed on: 11/08/2021).

[3] ABI Research. Small Unmanned Aerial Systems Market Exceeds US$8.4 Billion by 2019, Dominated by the Commercial Sector and Driven by Commercial Applications. www.abiresearch.com/press/small-unmanned-arial-systems-market-exceeds-us84-b (Accessed on: 12/08/2021).

[4] Tesla, N. The Wonder World To Be Created By Electricity, 1915.

[5] Draper, M. *Sitting Ducks & Peeping Toms: Targets, Drones and UAVs in British Military Service Since 1917*. Air Britain Historians Ltd, 2011.

[6] Tesla, N. Method of and apparatus for controlling mechanism of moving vessels or vehicles (US613809 A), 1898.

[7] Yu, T. *The development of a hybrid simulation modelling approach based on agents and discrete-event modelling*. PhD thesis, University of Southampton, 2008.

[8] Neumann, P.G. Modeling and simulation - inside risks. *Communications of the ACM*, 36(4):124, 1993.

[9] Wiese, P. and John, P. *Engineering Design in the Multi-Discipline Era, A Systems Approach*. Wiley, 2003.

[10] Sheard, S. and Mostashari, A. A Complexity Typology for Systems Engineering. *International Symposium of the International Council on Systems Engineering*, 20(Schwager):933–945, 2010.

[11] Collopy, P.D. and Hollingsworth, P.M. Value-Driven Design. *Journal of Aircraft*, 48(3):749–759, 2011.

[12] Zipline International Inc. Zipline. https://www.flyzipline.com/ (Accessed on 19/08/2022), 2022.

[13] Gavi the Vaccine Alliance. Rwanda launches world's first national drone delivery service powered by Zipline. https://www.gavi.org/news/media-room/rwanda-launches-worlds-first-national-drone-delivery-service-powered-zipline (Accessed on 12/10/2022), 2016.

[14] Maritime and Coastguard Agency. UK Second-Generation Search and Rescue Aviation programme (UKSAR2G). www.gov.uk/government/publications/second-generation-uk-search-and-rescue-aviation-programme-uksar2g (Accessed on 12/08/2021).

[15] Maritime and Coastguard Agency. Second Generation Search & Rescue Aviation Programme (UKSAR2G) Industry Day. https://www.ukhma.org/wp-content/uploads/2020/08/UKSAR2G_Industry_Day_Presentation.pdf (Accessed on 13/10/2022), 2020.

[16] Civil Aviation Authority. An introduction to unmanned aircraft systems. www.caa.co.uk/Consumers/Unmanned-aircraft/Our-role/An-introduction-to-unmanned-aircraft-systems (Accessed on: 12/08/2021).

[17] Civil Aviation Authority. CAP 722: Unmanned Aircraft System Operations in UK Airspace – Guidance. *Safety and Airspace Regulation Group*, CAP(Sixth Edition March 2015), 2015.

[18] Austin, R. *Unmanned Aircraft Systems - UAVS Design, Development and Deployment.* Wiley, 2010.

[19] Papageorgiou, E. *Value driven design of unmanned air systems.* PhD thesis, University of Southampton, 2016.

[20] Frampton, R. UAV Autonomy. *Defence Codex The Journal for Defence Engineering and Science*, 1(1):28 – 31, 2008.

[21] Northrop Grumman. RQ-4 Block 30 Global Hawk Datasheet. http://www.aurora.aero/wp-content/uploads/2015/09/Northrop-Grumman-Global-Hawk-Brochure.pdf (Accessed on: 12/08/2021).

[22] IAI. IAI Heron Datasheet. https://www.iai.co.il/p/heron (Accessed on: 12/08/2021).

[23] Thales. Watchkeeper Datasheet. www.thalesgroup.com/sites/default/files/database/d7/asset/document/3862_tha_watchkeeper_brochure_2013_.pdf (Accessed on: 12/08/2021), 2013.

[24] Ferraro, M., Lock, A., Scanlan, J., and Keane, A. Design and flight test of a civil unmanned aerial vehicle for maritime patrol: the use of 3D-printed structural components. *4th Aircraft Structural Design Conference*, 2014.

[25] Univeristy of Southampton. Southampton University Laser Sintered Aircraft. www.southampton.ac.uk/~decode/Sulsa_presentation.pdf (Accessed on: 12/08/2021).

[26] FLIR. Black Hornet PRS Datasheet. www.flir.com/globalassets/imported-assets/document/black-hornet-prs-spec-sheet.pdf (Accessed on: 12/08/2021).

[27] Civil Aviation Authority. CAP2012 Requirements for Flying in the Open Category (December 2020). http://publicapps.caa.co.uk/docs/33/CAP2012_EU_Drone_Rules_Factsheet_V7 7.pdf (Accessed on: 14/08/2021).

[28] Humphreys, K. and Wellman, P. *Basic cost engineering.* M Dekker, New York, 3rd, revis edition, 1996.

[29] Asiedu, Y. and Gu, P. Product life cycle cost analysis: State of the art review. *International Journal of Production Research*, 36(4):883–908, 1998.

[30] Ahmed, N.U. A design and implementation model for life cycle cost management system. *Information & Management*, 28:261–269, 1995.

[31] Moubray, J. *Reliability-centered maintenance.* Butterworth-Heinemann, 2nd edition, 1997.

[32] Lewis, E.E. *Introduction to reliability engineering.* Wiley, 2nd edition, 1996.

[33] Bartholomew-Biggs, M., Zuo, M.J., and Li, X. Modelling and optimizing sequential imperfect preventive maintenance. *Reliability Engineering and System Safety*, 94(1):53–62, 2009.

[34] Kijima, M. Some Results for Repairable Systems with General Repair. *Journal of Applied Probability*, 26(1):89–102, 1989.

[35] Schumann, B. *Aeronautical Life-Cycle Mission Modelling Framework for Conceptual Design.* PhD thesis, University of Southampton, 2014.

[36] Duquette, M. Effects-Level Models for UAV Simulation. In *AIAA Modeling and Simulation Technologies Conference*, pages 1–12, Reston, Virigina, 8 2009. American Institute of Aeronautics and Astronautics.

[37] Goerzen, C., Kong, Z., and Mettler, B. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 57(1-4):65–100, 2010.

[38] Ostler, J.N., Bowman, W.J., Snyder, D.O., and McLain, T.W. Performance Flight Testing of Small, Electric Powered Unmanned Aerial Vehicles. *International Journal of Micro Air Vehicles*, 1(3):155–171, 2009.

[39] Anderson, J.D.J. *Introduction to flight.* McGraw-Hill Education, New York, NY, eighth edition, 2016.

[40] Breguet, L. Aerodynamical Efficiency and the Reduction of Air Transport Costs. *Aeronautical journal (London, England : 1897)*, 26(140):307–320, 8 1922.

[41] Peckham, D.H. Range Performance in Cruising Flight. *Royal Aircraft Establishment*, TR-73164(March), 1974.

[42] Ferraro, M. *Value Driven Design of Additive Manufactured Unmanned Aircraft.* PhD thesis, University of Southampton, 2016.

[43] Retana, E.R. and Rodriguez-Cortes, H. Basic Small Fixed Wing Aircraft Sizing Optimizing Endurance. In *2007 4th International Conference on Electrical and Electronics Engineering*, pages 322–325. IEEE, 9 2007.

[44] Traub, L.W. Range and endurance estimates for battery-powered aircraft. *Journal of Aircraft*, 48(2):703–707, 2011.

[45] Peukert, W. About the Dependence of the Capacity of the Discharge Current Magnitude and Lead Acid Batteries. *Elektrotechnische Zeitschrift*, 20:287–288, 1897.

[46] Sun, Y.H., Jou, H.L., and Wu, J.C. Multilevel peukert equations based residual capacity estimation method for lead-acid battery. *2008 IEEE International Conference on Sustainable Energy Technologies, ICSET 2008*, pages 101–105, 2008.

[47] Traub, L.W. Validation of endurance estimates for battery powered UAVs. *Aeronautical Journal*, 117(1197):1155–1166, 2013.

[48] Boyd, C., Westcott, O., Ferraro, M., et al. CASCADE Open Aircraft Project: University of Southampton VTOL Drone Development. In *AIAA Scitech 2021 Forum*, pages 1–18. American Institute of Aeronautics and Astronautics, 1 2021.

[49] Doerffel, D. and Sharkh, S.A. A critical review of using the Peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries. *Journal of Power Sources*, 155(2):395–400, 4 2006.

[50] Soton UAV. Project OPSARS. www.sotonuav.uk/projects/index.html#opsars (Accessed on: 12/08/2021).

[51] Sandvik, K.B. and Lohne, K. The Rise of the Humanitarian Drone: Giving Content to an Emerging Concept. *Millennium: Journal of International Studies*, 43(1):145–164, 9 2014.

[52] Surendra, A. *Development of a Decision Support Framework for Systems Architecting in Aerospace Applications.* PhD thesis, University of Southampton, 2015.

[53] Leachtenauer, J.C. and Driggers, R.G. *Surveillance and Reconnaissance Imaging: Modeling and Performance Prediction*. Artech House, 2001.

[54] Gundlach, J. *Designing Unmanned Aircraft Systems: A Comprehensive Approach*. AIAA Education Series, 2012.

[55] Irvine, J.M. National image interpretability rating scales: overview and methodology. *Preceedings of the International Society for Optical Engineering (SPIE)*, 3128:93–103, 1997.

[56] Leachtenauer, J.C., Malila, W., Irvine, J., et al. General Image-Quality Equation: {GIQE}. *Applied Optics*, 36(32):8322–8328, 1997.

[57] Maria, A. Introduction to modeling and simulation. In *Proceedings of the 1997 Winter Simulation Conference*, pages 7–13, 1997.

[58] Gilbert, J.K., Boulter, C., and Rutherford, M. Models in explanations, Part 1: Horses for courses? *International Journal of Science Education*, 20(1):83–97, 1998.

[59] Law, A.M. and Kelton, W.D. *Simulation Modeling & Analysis*. McGraw-Hill, Singapore, second edition, 1991.

[60] Conway, R.W. Some tactical problems in digital simulation. *Management Science*, 10(1):47–62, 1963.

[61] Rubinstein, R.Y. and Kroese, D.P. *Simulation and the Monte Carlo Method*. Wiley, second edi edition, 2008.

[62] Nance, R.E. and Sargent, R.G. Electrical Engineering and Computer Science Perspectives on the Evolution of Simulation. *Operations Research*, 50(1):161–172, 2002.

[63] Shannon, R.E. Introduction to Simulation Languages. In *Proceedings of the 1977 Winter Simulation Conference*, pages 15–20, 1977.

[64] Borshchev, A. and Filippov, A. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. *22nd International Conference of the System Dynamics Society, 25-29 July 2004*, page 45, 2004.

[65] Jinks, S. *Integrating supply chain simulation, component geometry and unit cost estimation*. PhD thesis, University of Southampton, 2012.

[66] Forrester, J.W. Industrial Dynamics: A Major Breakthrough for Decision Makers. *Havard Business Review*, 36(4):37 – 66, 1958.

[67] Sajadi, S.M., Seyed Esfahani, M.M., and Sörensen, K. Production control in a failure-prone manufacturing network using discrete event simulation and automated response surface methodology. *The International Journal of Advanced Manufacturing Technology*, 53(1-4):35–46, 2011.

[68] Chen, Y., Mockus, L., Orcun, S., and Reklaitis, G.V. Simulation-optimization approach to clinical trial supply chain management with demand scenario forecast. *Computers and Chemical Engineering*, 40:82–96, 2012.

[69] Fujimoto, R. Parallel and distributed simulation. In *2015 Winter Simulation Conference (WSC)*, volume 1, pages 45–59. IEEE, 12 2015.

[70] Fujimoto, R.M. Parallel discrete event simulation. *Communications of the ACM*, 33(10):30–53, 10 1990.

[71] Jennings, N.R. On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296, 2000.

[72] Wooldridge, M. Agent-based software engineering. *IEE Proceedings Software Engineering.*, 144(1):26–37, 1997.

[73] Macal, C.M. Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, 10(2):144–156, 5 2016.

[74] Wooldridge, M. and Jennings, N.R. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 6 1995.

[75] Scheutz, M. and Logan, B. Affective vs. deliberative agent control. In *Proceedings of the AISB'01 symposium on emotion, cognition and affective computing.*, pages 1–10, 2001.

[76] Conte, R. and Paolucci, M. On agent-based modeling and computational social science. *Frontiers in Psychology*, 5(JUL):1–9, 7 2014.

[77] Irnich, S., Toth, P., and Vigo, D. The Family of Vehicle routing Problems. In *Vehicle routing : problems, methods, and applications*, MOS-SIAM series on optimization, chapter 1, pages 1–33. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), Philadelphia, Pennsylvania, second edi edition, 2014.

[78] Dantzig, G. and Ramser, J. The Truck Dispatching Problem. *Management Science*, 6(1):80–91, 1959.

[79] Baldacci, R., Battarra, M., and Vigo, D. Routing a Heterogeneous Fleet of Vehicles. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 3–27. Springer US, Boston, MA, 2008.

[80] Irnich, S., Schneider, M., and Vigo, D. Four Variants of the Vehicle Routing Problem. In *Vehicle Routing*, chapter 9, pages 241–271. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), second edi edition, 2014.

[81] Ojeda Rios, B.H., Xavier, E.C., Miyazawa, F.K., et al. Recent dynamic vehicle routing problems: A survey. *Computers & Industrial Engineering*, 160:107604, 2021.

[82] Robinson, S. *Simulation: the practice of model development and use*. John Wiley & Sons, Ltd, 2004.

[83] Innis, G. and Rexstad, E. Simulation model simplification techniques. *Simulation*, 41:7–15, 1983.

[84] Chwif, L., Barretto, M., and Paul, R. On simulation model complexity. In *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, volume 1, pages 449–455. IEEE, 2000.

[85] Costanza, R. and Sklar, F.H. Articulation, accuracy and effectiveness of mathematical models: A review of freshwater wetland applications. *Ecological Modelling*, 27(1-2):45–68, 1985.

[86] Lobão, E.d.C. and Porto, A.J.V. Proposta para sistematização de estudos de simulação. *Encontro Nacional De Engenharia De Produção*, 1997.

[87] Fulton, E.A., Smith, A.D.M., and Johnson, C.R. Effect of complexity on marine ecosystem models. *Marine Ecology Progress Series*, 253:1–16, 2003.

[88] Amiry, A.P. The simulation of information flow in a steelmaking plant. *Digital Simulation in Operational Research*, pages 347–356, 1965.

[89] Hurrion, R.D. *The design, use and required facilities of an interactive visual computer simulation language to explore production planning problems*. PhD thesis, University of London, 1976.

[90] Banks, J. and Gibson, R.R. The ABCs of Simulation Practice. *Analytics Magazine*, pages 16 – 21, 2009.

[91] Ross, A., O'Neill, G., Hastings, D., and Rhodes, D. Aligning Perspectives and Methods for Value-Driven Design. *AIAA SPACE 2010 Conference & Exposition*, pages 1–30, 2010.

[92] Collopy, P. Aerospace System Value Models: A Survey and Observations. In *AIAA SPACE 2009 Conference & Exposition*, pages 1–18, Reston, Virigina, 9 2009. American Institute of Aeronautics and Astronautics.

[93] Brown, O., Eremenko, P., and Collopy, P. Value-Centric Design Methodologies for Fractionated Spacecraft: Progress Summary from Phase I of the DARPA System F6 Program. In *AIAA SPACE 2009 Conference & Exposition*, Reston, Virigina, 9 2009. American Institute of Aeronautics and Astronautics.

[94] Peoples, R.E. and Willcox, K.E. Value-Based Multidisciplinary Optimization for Commercial Aircraft Design and Business Risk Assessment. *Journal of Aircraft*, 43(4):913–921, 2006.

[95] Price, M., Raghunathan, S., and Curran, R. An integrated systems engineering approach to aircraft design. *Progress in Aerospace Sciences*, 42(4):331–376, 2006.

[96] Collopy, P.D. A System for Values, Communication, and Leadership in Product Design. *International Powered Lift Conference Proceedings, Warrendale, PA*, pages 95–98, 1997.

[97] Cheung, J., Scanlan, J., Wong, J., et al. Application of Value-Driven Design to Commercial Aeroengine Systems. *Journal of Aircraft*, 49(3):688–702, 2012.

[98] Dahlgren, J.W. Real Options and Value Driven Design in Spiral Development. *INCOSE International Symposium*, 16(1):1308–1317, 7 2006.

[99] Arrow, K.J. *Social Choice and Individual Values*. Yale University Press, 2nd edition, 1963.

[100] Papageorgiou, E., Eres, M.H., and Scanlan, J. Value modelling for multi-stakeholder and multi-objective optimisation in engineering design. *Journal of Engineering Design*, 27(10):697–724, 2016.

[101] Lai, V., Wong, B., and Cheung, W. Group decision making in a multiple criteria environment: a case using the AHP in software selection. *European Journal of Operational research*, 137:134–144, 2002.

[102] Sen, P. and Yang, J.B. *Multiple Criteria Decision Support in Engineering Design*. Springer London, London, 1998.

[103] Saaty, T.L. How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26, 1990.

[104] Dyer, J.S. Remarks on the Analytic Hierarchy Process. *Management Science*, 36(3):249 – 258, 1990.

[105] Velasquez, M. and Hester, P.T. An Analysis of Multi-Criteria Decision Making Methods. *International Journal of Operations Research*, 10(2):56–66, 2013.

[106] Pérez, J. Some Comments on Saaty's AHP. *Management Science*, 41(6):1091–1095, 1995.

[107] Von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behaviour*. Princeton University Press, 1944.

[108] Keeney, R.L. and Raiffa, H. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.

[109] Ross, A.M., Hastings, D.E., Warmkessel, J.M., and Diller, N.P. Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design. *Journal of Spacecraft and Rockets*, 41(1):20–28, 2004.

[110] van Calker, K.J., Berentsen, P.B.M., Romero, C., et al. Development and application of a multi-attribute sustainability function for Dutch dairy farming systems. *Ecological Economics*, 57(4):640–658, 2006.

[111] Nurminen, J.K., Karonen, O., and Hatonen, K. What makes expert systems survive over 10 years-empirical evaluation of several engineering applications. *Expert Systems with Applications*, 24(3):199–211, 2003.

[112] UK Government Legislation. Regulation (EU) 2018/1139 of the European Parliment. https://www.legislation.gov.uk/eur/2018/1139/contents# (Accessed 08/09/2021), 2018.

[113] Pham, H. and Lai, C.D. On recent generalizations of the Weibull distribution. *IEEE Transactions on Reliability*, 56(3):454–458, 2007.

[114] Lai, C.D., Xie, M., and Murthy, D.N. A modified Weibull distribution. *IEEE Transactions on Reliability*, 52(1):33–37, 2003.

[115] Bidaoui, H., Abbassi, I.E., Bouardi, A.E., and Darcherif, A. Wind Speed Data Analysis Using Weibull and Rayleigh Distribution Functions, Case Study: Five Cities Northern Morocco. *Procedia Manufacturing*, 32:786–793, 2019.

[116] Chen, H., Birkelund, Y., Anfinsen, S.N., et al. Assessing probabilistic modelling for wind speed from numerical weather prediction model and observation in the Arctic. *Scientific Reports*, 11(1):7613, 12 2021.

[117] Wang, L., Liu, J., and Qian, F. Frequency Distribution Model of Wind Speed Based on the Exponential Polynomial for Wind Farms. *Sustainability*, 11(3):665, 1 2019.

[118] Tennekes, H. The Logarithmic Wind Profile. *Journal of the Atmospheric Sciences*, 30(2):234–238, 3 1973.

[119] Abar, S., Theodoropoulos, G.K., Lemarinier, P., and O'Hare, G.M. Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, 2017.

[120] AGI. AGI Engineering Tools Analysis. www.agi.com/products/stk (Accessed on: 12/08/2021).

[121] AGI. AGI STK Analysis Add-on Modules. help.agi.com/stk/index.htm#analysisModules.htm (Accessed on: 12/08/2021).

[122] Presagis. Presagis STAGE product datasheet. www.presagis.com/workspace/uploads/files/11172019_pres_print_brochure_stage_v2.pdf (Accessed on: 12-08-2021).

[123] Ternion. FLAMES Simulation Framework. www.ternion.com/print/FLAMES-Constructive-Simulation-Framework-Brochure.pdf (Accessed on: 12/08/2021).

[124] Cassidy, P.F., Gatzke, T.D., and Vaporean, C.N. Integrating Synthesis and Simulation for Conceptual Design. *46th AIAA Aerospace Sciences Meeting and Exhibit*, pages 1443 – 1451, 2008.

[125] NASA. Welcome to NASA World Wind. https://worldwind.arc.nasa.gov (Accessed on: 12/08/2021).

[126] Sharif, K. and Salah, F. NASA World Weather. https://github.com/WorldWindLabs/WorldWeather (Accessed on: 12/08/2021).

[127] Militão, G. and Chang, B. Quake Hunter: Earthquake Activity Visualizer, 2016.

[128] Del Castillo, M., Stewart, B., and Semenenko, J. SpaceBirds. https://github.com/WorldWindLabs/SpaceBirds (Accessed on 12/08/2021), 2020.

[129] Bölstler, F. Passenger Flow Simulation at Frankfurt Airport. www.anylogic.com/upload/case study/Fraport_Success_Story.pdf (Accessed on: 12/08/2021).

[130] AnyLogic and Evans & Peck. Rail Yard Capacity Modeling. www.anylogic.com/rail-yard-capacity-modeling (Accessed on: 12/08/2021).

[131] Grignard, A., Taillandier, P., Gaudou, B., et al. GAMA 1.6 : Advancing the art of complex agent-based modeling and simulation. *PRIMA 2013: Principles and Practice of Multi-Agent Systems*, pages 117–131, 2013.

[132] Imagine That Inc. ExtendSim Overview. https://www.extendsim.com (Accessed on: 12/08/2021).

[133] Tewoldeberhan, T.W., Verbraeck, A., Valentin, E., and Bardonnet, G. An Evaluation and Selection Methodology for Discrete Event Simulation Software. *Proceedings of the 2002 Winter Simulation Conference*, pages 67–75, 2002.

[134] Team SimPy. SimpPy. simpy.readthedocs.io (Accessed on: 12/08/2021).

[135] Python Software Foundation. Python Language Reference, version 2.7. www.python.org (Accessed on: 12/08/2021).

[136] Royal National Lifeboat Institution. RNLI Open Data. https://rnli.org/about-us/our-research/rnli-open-data (Accessed on 12/08/2021).

[137] Schellenberg, B., Richardson, T., Watson, M., et al. Remote sensing and identification of volcanic plumes using fixed-wing UAVs over Volcán de Fuego, Guatemala. *Journal of Field Robotics*, 36(7):1192–1211, 2019.

[138] Weber, L.M., Saelens, W., Cannoodt, R., et al. Essential guidelines for computational method benchmarking. *Genome Biology*, 20(1):125, 2019.

[139] Mangul, S., Martin, L.S., Hill, B.L., et al. Systematic benchmarking of omics computational tools. *Nature Communications*, 10(1):1393, 2019.

[140] Boulesteix, A.L., Lauer, S., and Eugster, M.J. A Plea for Neutral Comparison Studies in Computational Sciences. *PLoS ONE*, 8(4), 2013.

[141] Boulesteix, A.L., Binder, H., Abrahamowicz, M., and Sauerbrei, W. On the necessity and design of studies comparing statistical methods. *Biometrical Journal*, 60(1):216–218, 2018.

[142] Norel, R., Rice, J.J., and Stolovitzky, G. The self-assessment trap: Can we all be better than average? *Molecular Systems Biology*, 7(537):1–2, 2011.

[143] Google. OR-Tools. \url{https://developers.google.com/optimization} (Accessed on: 22/10/2022).

[144] Hoos, H.H. and Stützle, T. Stochastic Local Search, Foundations and Applications. In Hoos, H.H. and Stützle, T.B.T.S.L.S., editors, *The Morgan Kaufmann Series in Artificial Intelligence*, chapter 1. Introdu, pages 13–59. Morgan Kaufmann, San Francisco, 2005.

[145] Nowatzki, T., Ferris, M.C., Sankaralingam, K., et al. *Optimization and mathematical modeling in computer architecture*. Springer, Cham, Switzerland, 2014.